Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра
(назва освітнього ступеня)

на тему: Практичні аспекти дослідження стійкості S-блоків до диференціального криптоаналізу

Виконав: студент 6 курсу, групи СБм-61
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

(підпис) Ярема О.М. (прізвище та ініціали)

Керівник (підпис) Загородна Н.В. (прізвище та ініціали)

Нормоконтроль (підпис) Лечаченко Т.А. (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
« » 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістра
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

студенту Яремі Олегу Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Практичні аспекти дослідження стійкості S-блоків до диференціального криптоаналізу

Керівник роботи к.т.н., доц. Загородна Наталія Володимирівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*16*» *листопада* 2023 року № *4/7-1061*

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ, 1 Аналіз предметної області та теоретичних відомостей, 1.1 Базові поняття предметної області, 1.2 Огляд симетричних шифрів, 1.3 Українські стандарти блокових шифрів, 1.4 Блоки підстановок, 1.5 Моделі та типи відомих атак на блочні шифри, 1.6 Постановка задачі, 2 Програмна реалізація алгоритмів для генерації S-блоків та диференціального криптоаналізу, 2.1 Методи генерації S-блоків, 2.2 Опис алгоритму диференціального криптоаналізу, 2.3 Обгрунтування вибору мови програмування та середовища розробки програмного забезпечення, 2.4 Розробка програмного забезпечення для генерації S-блоків, 2.5 Реалізація програмного забезпечення для диференціального криптоаналізу, 3 Експериментальне дослідження стійкості S-блоків, 3.1 Методика проведення дослідження, 3.2 Аналіз результатів експериментальних досліджень, 4 Охорона праці та безпека в надзвичайних ситуаціях, 4.1 Охорона праці, 4.2 Безпека в надзвичайних ситуаціях, Висновки, Список джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Стручок В.С., старший викладач		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Опрацювання завдання		
2	Аналіз джерел та огляд літератури		
3	Написання 1-го розділу		
4	Розробка програмного забезпечення		
5	Написання 2-го розділу		
6	Проведення експериментальних досліджень		
7	Написання 3-го розділу		
8	Опрацювання питань 4-го розділу		
9	Оформлення роботи		
10	Нормоконтроль		
11	Перевірка на плагіат		
12	Попередній захист		
13	Захист		

Студент _____ Ярема О.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Практичні аспекти дослідження стійкості S-блоків до диференціального криптоаналізу // Кваліфікаційна робота освітнього рівня «Магістр» // Ярема Олег Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2023 // С. 68, рис. – 13, табл. – 1, додат. – 2, бібліогр. – 23.

Ключові слова: S-БЛОК, КРИПТОЛОГІЯ, БЛОКОВИЙ ШИФР, ТАБЛИЦЯ ПІДСТАНОВКИ, ДИФЕРЕНЦІАЛЬНИЙ КРИПТОАНАЛІЗ, ДИФЕРЕНЦІАЛЬНА РІВНОМІРНІСТЬ.

В кваліфікаційній роботі виконано дослідження практичних аспектів стійкості S-блоків до диференціального криптоаналізу. Для цього було реалізовано програмне забезпечення для генерації S-блоків та проведення їх диференціального аналізу на основі використання згенерованих S-блоків в експериментальному одноразундовому блоковому шифрі. Проведено аналіз результатів диференціального криптоаналізу згенерованих S-блоків з різними значеннями криптографічних властивостей.

ANNOTATION

Practical Aspects of Research on Resistance of S-Blocks to Differential Cryptanalysis // Qualification paper of the educational level “Master” // Yarema Oleh // Ternopil Ivan Puluj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Cybersecurity, SBm-61 group // Ternopil, 2023 // P. 68, tables – 1, fig. – 13, annexes. – 2, references – 23.

Keywords: S-BOX, CRYPTOLOGY, BLOCK CIPHER, SUBSTITUTION TABLE, DIFFERENTIAL CRYPTANALYSIS, DIFFERENTIAL UNIFORMITY.

In this paper the practical aspects of S-boxes' resistance to differential cryptanalysis were studied. For this purpose, software was implemented to generate S-boxes and conduct their differential analysis based on the use of the generated S-boxes in an experimental single-round block cipher. The results of the differential cryptanalysis of the generated S-boxes with different values of cryptographic properties are analysed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БСШ — блоковий симетричний шифр

ІКС — інформаційно-комунікаційна система

ООП — об'єктно-орієнтовне програмування, об’єктна парадигма програмування

ТДР (DDT) — таблиця диференціального розподілу, англ. difference distribution table

AES (Advanced Encryption Standard) — симетричний блоковий шифр, стандарт шифрування США, інша назва — Rijndael

BLOWFISH — симетричний блоковий шифр, побудований на швидких і простих операціях

DES (Data Encryption Standard) — симетричний блоковий шифр, був стандартом США з симетричного шифрування

DPA — диференціальна атака на енергоспоживання, більш статистичний різновид атаки на енергоспоживання

IDE (integrated development environment) — інтегроване середовище розробки програмного забезпечення

IDEA (International Data Encryption Algorithm) — симетричний блоковий алгоритм шифрування даних

NIST (National Institute of Standards and Technology) — Національний інститут стандартів і технології США

SCA — атака сторонніми каналами, спрямована на практичну реалізацію криптосистеми

SPA — атака на енергоспоживання, спрямована на дослідження затрат електроенергії криптосистемою

SPN (substitution–permutation network) — ряд операцій заміни та перестановок, що використовуються в блокових шифрах

S-блок (S-box) — блок підстановки в блоковому симетричному шифрі, від англ. substitutional box

TRIPLEDES — симетричний блоковий шифр, похідний від DES

XOR (Exclusive OR) — логічна та бітова операція виключної диз'юнкції, інша назва - додавання за модулем 2

ВСТУП

S-блоки, також відомі як блоки підстановок, є фундаментальними компонентами багатьох симетричних блочних криптографічних алгоритмів. Вони відіграють вирішальну роль у досягненні нелінійних властивостей, необхідних для протистояння криптоаналізу, реалізуючи в шифрах властивість запутування можливих лінійних чи статистичних залежностей.

Диференціальний криптоаналіз, як один з основних методів атаки, використовує слабкі місця в цих властивостях для відновлення секретного ключа шифру. Тому забезпечення стійкості S-блоків до безпосередньо диференціального криптоаналізу має першорядне значення в питанні стійкості та безпеки блокових шифрів. І саме це є одним з основних питань для розробки нових блокових шифрів.

Ця робота заглиблюється в практичні аспекти дослідження стійкості S-блоків до диференціального криптоаналізу. Ми досліджуємо існуючі та нові методології аналізу та оцінки їх безпеки, зосереджуючись на практичних міркуваннях та реальних застосуваннях.

Було проведено огляд предметної області криптології, шифрів та диференціального криптоаналізу, включаючи його основні принципи та різні стратегії атак для розуміння конкретних проблем і вразливостей S-блоків.

Далі в роботі проведено аналіз методів генерації S-блоків, їх диференціального криптоаналізу та розроблене програмне забезпечення, яке реалізує практичні алгоритми, які будуть використовуватися для проведення експериментального дослідження.

Використовуючи розроблене програмне забезпечення, досліджено стійкість до диференціального криптоаналізу згенерованих S-блоків з різним рівнем властивості диференціальної рівномірності.

Виходячи за межі теоретичного аналізу, в роботі наведено практичні міркування щодо проектування та впровадження S-блоків із сильною диференціальною стійкістю та визначено перспективні напрямки досліджень в області дизайну S-блоків і диференціального криптоаналізу. Обговорюємо

потенційні виклики та можливості для подальшого підвищення безпеки криптографічних систем.

Надаючи окремий погляд на дослідження практичних аспектів стійкості S-блоків, ця робота має на меті зробити значний внесок у розвиток криптографічної теорії та практики.

Метою даної роботи є практичний аналіз стійкості S-блоків з різним ступенем диференціальної рівномірності до диференціального криптоаналізу з метою визначення оптимального рівня диференціальної рівномірності.

Щоб досягнути поставленої мети, потрібно вирішити такі задачі:

– Дослідити та проаналізувати джерела предметної області дослідження.

– Реалізувати програмне забезпечення для генерування S-блоків з різним ступенем диференціальної рівномірності.

– Реалізувати програмне забезпечення для диференціального криптоаналізу S-блоків в межах однораундового блокового шифру.

– Провести диференціальний криптоаналіз для множини S-блоків з різними ступенями диференціальної рівномірності.

– Визначити перспективні напрямки подальшого дослідження S-блоків на основі даного дослідження.

Об’єкт дослідження – S-блоки блокових шифрів.

Предметом дослідження є стійкість S-блоків до диференціального криптоаналізу.

У роботі використовувались наступні методи дослідження: експеримент та вимірювання (програмна реалізація алгоритмів та вимірювання кількісних значень критеріїв ефективності), аналіз та синтез (аналіз отриманих кількісних показників критеріїв та виконання порівняльного аналізу на основі синтезу отриманої інформації).

Науковою новизною є дослідження стійкості S-блоків до диференціального криптоаналізу з врахуванням ступеня диференціальної рівномірності.

Практичною цінністю одержаних результатів є можливість використання дослідження для оптимізації генерації стійких до диференціального криптоаналізу S-блоків, які використовуватимуться в нових блокових шифрах.

Основні положення і результати роботи були представлені на XI науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 13-14 грудня 2023 р.).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕОРЕТИЧНИХ ВІДОМОСТЕЙ

1.1 Базові поняття предметної області

Канал зв'язку - це будь-який фізичний засіб для передачі інформації. Безпечний канал характеризується тим, що він несприйнятливий до підслуховування та перешкод. Для порівняння, незахищений канал не гарантує ні конфіденційності, ні цілісності. Що стосується повідомлень, то осмислена і зрозуміла інформація в певних умовах називається відкритим текстом, а її прихована форма - шифрованим текстом.

Як зазначає Борст у [9], "...досягнення інформаційної безпеки передбачає перенесення довіри з ненадійних компонентів на компоненти, яким можна довіряти. Це може бути може бути ефективно досягнуто за допомогою криптографічних методів".

Цілі інформаційної безпеки включають: конфіденційність, цілісність даних, автентифікацію об'єктів і повідомлень, підпис, авторизація, валідація, контроль доступу, сертифікація, відмітка часу, свідчення, отримання, підтвердження правf власності, анонімність, неспростування та відкликання.

Криптологія включає в себе вивчення математичних методів, пов'язаних з такими аспектами інформаційної безпеки, як конфіденційність, цілісність даних, ідентифікація суб'єкта та автентифікація сутності та походження даних. Криптологія складається з двох взаємодоповнюючих галузей: криптографії та криптоаналізу.

Всі цілі інформаційної безпеки з точки зору криптографії можуть бути досягнуті за допомогою системи, що дотримується наступних властивостей:

– Конфіденційність: це властивість, яка спрямована на збереження в таємниці змісту інформації для всіх сторін, за винятком тих, хто має право доступу до неї.

– Цілісність даних: властивість, яка спрямована на запобігання несанкціонованій зміні даних. Вона спрямована на виявлення незаконних маніпуляцій з даними, таких як вставка, видалення та підміну даних.

– Автентифікація: може бути розділена на автентифікацію суб'єкта, яка пов'язана з ідентифікацією сторін, що вступають у сеанс зв'язку; та автентифікацію джерела даних, яка надає ідентифікаційну інформацію про відправника або джерела даних.

– Відмова від заперечення: послуга, яка не дозволяє суб'єкту заперечувати попередні зобов'язання або дії.

Для досягнення цілей інформаційної безпеки використовується набір криптографічних інструментів або криптографічних примітивів. Прикладами примітивів є алгоритми шифрування, хеш-функції та схеми цифрового підпису [23]. Ці примітиви можуть бути використані як будівельні блоки для інших примітивів або криптосистем (шифрів).

У сфері безпечних каналів зв’язку шифруванням інформації керують дві фундаментальні парадигми: симетрична та асиметрична криптографія (див. рис. 1.1). Обидві методології використовують математичні перетворення, щоб зробити дані нерозбірливими, але їхні принципи управління ключами та роботи суттєво відрізняються.

Симетрична криптографія дотримується парадигми загального ключа. Єдиний секретний ключ полегшує як шифрування, так і розшифрування. Цей підхід може похвалитися винятковою обчислювальною ефективністю, що робить його ідеальним для шифрування великих обсягів даних. Однак, безпечне розповсюдження та обслуговування ключів набуває першочергового значення, оскільки будь-яка несанкціонована особа, яка отримує ключ, компрометує всю систему.

Асиметрична криптографія, навпаки, використовує пари ключів. Кожен учасник володіє загальнодоступним відкритим ключем і приватним ключем, який ретельно охороняється. Відкритий ключ слугує шлюзом для шифрування, а закритий - оракулом для дешифрування. Ця парадигма усуває потребу в

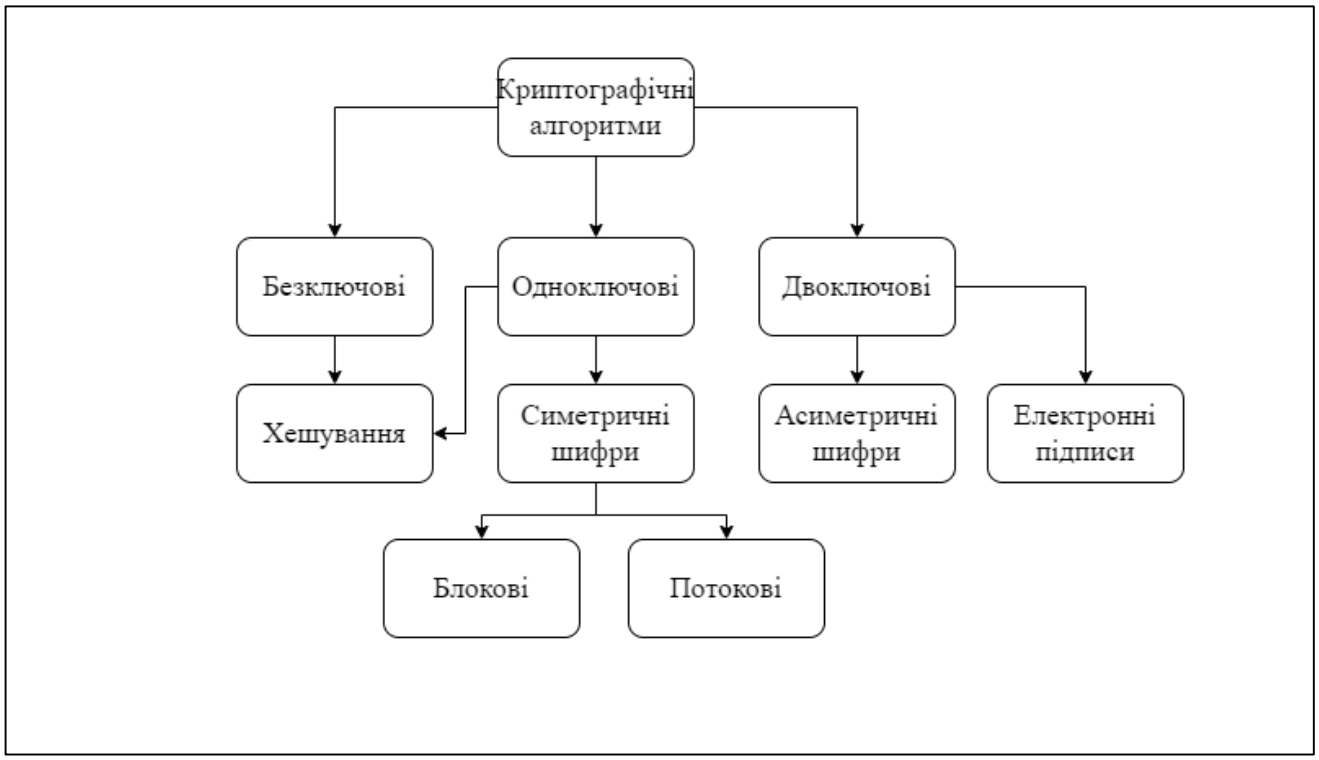

Рисунок 1.1 – Класифікація криптографічних алгоритмів

безпечному обміні ключами, підвищуючи безпеку комунікації та уможливлюючи використання цифрових підписів. Однак асиметричні алгоритми несуть значні обчислювальні витрати, що робить їх менш придатними для шифрування великих обсягів даних.

Вибір між симетричною та асиметричною криптографією залежить від конкретних вимог безпеки та операційного контексту. Якщо швидкість і ефективність мають першорядне значення, а безпечний розподіл ключів є можливим, симетрична криптографія є переконливим вибором. Однак для додатків, що вимагають підвищеної безпеки, цифрових підписів і безпечного спілкування без спільних секретів, асиметрична криптографія має перевагу. Розуміння нюансів цих криптографічних парадигм дає людям можливість захистити інформацію в цифровому світі, повному потенційних вразливостей.

Симетричні алгоритми шифрування дуже ефективні при обробці великих обсягів інформації і менш трудомісткі, ніж асиметричні алгоритми шифрування. Існує два типи симетричних алгоритмів шифрування: потокові шифри та блокові

шифри, які забезпечують побітове та блокове шифрування відповідно. І саме блокові симетричні шифри до сьогодні залишаються одними з найпопулярніших

1.2 Огляд симетричних шифрів

Існують різні блокові алгоритми з симетричним ключем, такі як DES, TRIPLEDES, AES і BLOWFISH.

DES був першим стандартом шифрування, який був опублікованим NIST (Національним інститутом стандартів Він був розроблений компанією IBM на основі на основі шифру Люцифера. Спочатку 56 біт ключа вибираються з початкових 64 шляхом перестановки. Решта вісім бітів або відкидаються, або використовуються як біти перевірки на парність. Потім 56 біт діляться на дві 28-бітові половини, кожна з яких потім обробляється окремо. У наступних раундах обидві половини повертаються вліво на один або два біти, а потім 48 додаткових бітів вибираються шляхом перестановки, 24 біти з лівої половини і 24 з правої. Розклад ключа для розшифрування аналогічна, підключі розташовані у зворотному порядку порівняно з шифруванням. Шифр виявився вразливим до диференціального криптоаналізу, незважаючи на спроби покращити його S-блоки [17].

AES - це блоковий шифр з симетричним ключем, опублікований Національним інститутом стандартів і технологій (NIST) в грудні 2001 р. Базові характеристики його версій вказані в таблиці 1.1.

Таблиця 1.1 - Характеристики різних типів AES

Тип	Кількість блоків ключів	Кількість блоків повідомлень	Кількість раундів
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Кожен раунд обробки складається з чотирьох кроків:

– Підстановка байтів: Використовує S-блок для виконання заміни байта байт за байтом у блоці.

– Перестановка рядків: Проста перестановка.

– Перемішати стовпчик: Метод заміни, де дані в кожному стовпчику зі зсувного рядка кожному стовпчику зі зсувного рядка множаться на матрицю алгоритму.

– Додати ключ округлення: Ключ для раунду обробки додається до даних за допомогою операції XOR.

У криптографії TRIPLE DES є загальною назва для потрійного алгоритму шифрування даних (Triple Data Encryption Algorithm) шифрування, який застосовує стандартний алгоритм шифрування даних алгоритм шифрування Data Encryption Standard тричі до кожного блоку даних. Початковий розмір ключа шифру DES в 56 біт був загалом достатнім, коли цей алгоритм був розроблений, але наявність зростаючих обчислювальних потужностей зробила атаки грубої сили доступним варіантом зламу. Потрійний DES забезпечує відносно простий метод збільшення розміру ключа DES для захисту від таких атак. Для цього потрібно використати три 64-бітних ключа, при загальній довжині ключа 192 біта. У потрійному DES дані шифруються за допомогою першого ключа, розшифровуються за допомогою другого ключа, і, нарешті, зашифровуються третім ключем. Triple DES працює втричі повільніше, ніж DES, але він набагато безпечніший. Процедура дешифрування така ж, як і процедура шифрування, за винятком того, що вона виконується в зворотному порядку.

Blowfish - це симетричний блоковий шифр, який може бути ефективно використовувати для шифрування і захисту даних. Він використовує ключ змінної довжини, від 32 біт до 448 біт, що робить його ідеальним для захисту даних. Blowfish був розроблений в 1993 році Брюсом Шнайером як швидка, безкоштовна альтернатива існуючим алгоритмам шифрування. Алгоритм Blowfish - це мережа Фейстеля, яка повторює просту функцію шифрування 16 разів. Розмір блоку

становить 64 біти, а ключ може мати будь-яку довжину до 448 біт. Це значно швидший за більшість алгоритмів шифрування при реалізації на 32-бітних мікропроцесорах з великими кешами даних. Алгоритм ділиться на дві частини: частини розширення ключа і частини, яка шифрує дані. Подовження ключа перетворює його у кілька масивів підключів довжиною в 4168 байт.

1.3 Українські стандарти блокових шифрів

У 2009 році Президент України затвердив Доктрину інформаційної безпеки України. Ця доктрина визначає загрози для інформаційної безпеки України, включаючи несанкціонований доступ до інформаційних ресурсів через використання іноземних інформаційних технологій.

Щоб забезпечити інформаційну безпеку України, Адміністрація Держспецзв'язку ініціювала розробку власного криптографічного алгоритму блокового симетричного шифру (БСШ). Прийняття національного стандарту БСШ повинно сприяти підвищенню рівня інформаційної безпеки України.

Впровадження БСШ в Україні забезпечило необхідний захист конфіденційності та цілісності інформації та інформаційних ресурсів. Це дозволило захистити державні секрети та інші важливі дані від несанкціонованого доступу, а також комерційні дані від конкурентів і особисті дані злочинців. Крім того, використання БСШ дозволило захистити обробку інформації з підвищеною швидкодією в ІКС.

Прийняття національного стандарту БСШ є одним із важливих кроків у зміцненні інформаційної безпеки України. Він допоміг захистити країну від кібератак і дозволив людям і організаціям отримати доступ до ресурсів і інформації, які вони потребують.

Національний конкурс на кращий проект національного стандарту БСШ, який розпочався 15 жовтня 2006 року та закінчився в травні 2010 року, є важливою подією для України. Він дозволив визначити найкращий проект національного стандарту БСШ, який відповідає вимогам інформаційної безпеки України. Чотири

кандидати (Калина, Лабіринт, ADE та Мухомор) подано на конкурс ініціативно [20]. По суті, обидва шифри, Калина та ADE, є тією чи іншою модифікацією шифру зі структурою SPN, ланцюгом Файстеля або структурою IDEA. Багато видань також містять специфікації та результати досліджень цих шифрів. Розробка та освоєння науково-методичного забезпечення для синтезу та аналізу БСШ, а також виконання складних завдань щодо розробки та дослідження перспективних БСШ є результатами конкурсу. Уже було використано цю інформацію для впровадження в Україні міжнародного стандарту ISO/IEC 18033–3, а також інших стандартів. У національному конкурсі вимоги до перспективного БСШ відрізняються від Nessie тим, що він встановлює надвисокий рівень безпеки (гарантій), коли довжина ключа перевищує 256 бітів і довжина ключа перевищує 512 бітів. Шифри SHACAL-2, Калина, Мухомор і Threefish, які мають довжину блока 256 бітів або більше, а довжину ключа 512 бітів, вважаються БСШ надвисокого рівня стійкості.

Дослідження принципів синтезу симетричних криптоперетворень показали, що нині необхідно виділити три методологічні підходи для побудови перспективних БСШ. Вони фактично є кандидатами на стандарт БСШ Європейської програми NESSIE. Перша пов’язана з використанням структур SPN. Загальна структура включає квадратний тип SPN і байт-орієнтований шифр. Такі структури послужили основою для розробки та визнання БСШ Rijndael і його звуженої версії AES (FIPS-197), яка була побудована на основі шифру Square, попередньої розробки авторів. Цей напрям був достатньо досліджений, і результати показали, що «Калина» є кандидатом на національний стандарт БСШ.У процесі дослідження були виявлені БСШ, які мають подібну структуру до IDEA. Європейський стандарт IDEA, як відомо, пройшов тривалий шлях випробування та все ще гарантує задекларований рівень стійкості.

На початку 2000-х років Паскаль Юнод і Серж Воденей запропонували FOX як удосконалену БСШ. Алгоритм IDEA NXT, який раніше називався FOX, був розроблений Паскаль Юнод і Серж Воденей в лабораторії EPFL. Проект був розроблений з 2001 по 2003 рік і був випущений в 2003 році під назвою FOX. Компанія MediaCrypt оголосила про нього під назвою IDEA NXT у травні 2005

року. Алгоритм IDEA є нащадком, який використовує розширену схему Лея-Массея. IDEA NXT відомий своєю стійкістю до криптоаналізу. Він належить швейцарській компанії MediaCrypt, яка володіє правами на поширення IDEA та патентами на IDEA NXT. Сім'я модифікацій шифру IDEA NXT складається з семи різних розмірів блоків і ключів. Стандарт NXT64 складається з 64 біт, 128 біт і 12 раундів, а стандарт NXT128 складається з 128 біт і 256 біт і має 12 раундів. Крім того, можуть бути створені версії Standard, які можуть мати розмір ключа від 0 до 256 бітів і кількість раундів від 2 до 255. Крім того, вони можуть використовуватися як заміна стандартної таблиці для завантаження індивідуальних таблиць, таких як сбокс або матриця перестановок.

Третій методологічний підхід до проектування блокових симетричних шифрів (БСШ) ґрунтується на добре випробуваній фейстельподібній схемі. Ця схема реалізована в таких стандартах БСШ, як DES, DEA, TDEA, ГОСТ 28147–89, MISTY1 та Cammellia. Сьогодні стандарти БСШ з фейстельподібною структурою все ще широко застосовуються і не втратили перспективи розвитку.

Аналіз принципів проектування сучасних шифрів показав, що однією з найпоширеніших і найпотужніших сучасних концепцій є стратегія широкого сліду. Ця стратегія будується на основі матричного множення в полях. Розробники AES використали цю стратегію, що дозволило їм обґрунтувати значення окремих показників ефективності БСШ, зокрема отримати просту специфікацію шифру, яку легко аналізувати за допомогою прозорого та надійного математичного апарату [2].

Через використання додавання ключів за модулем 2^{32}, "Калина" не є шифром Маркова, тому традиційний метод оцінки стійкості до ДК не може бути застосований безпосередньо. Традиційний метод базується на обчисленні суми кількості активних S-блоків у кожному раунді в процесі шифрування. Маючи це значення та максимальну ймовірність різницевого перетворення S-блоків, можна оцінити верхню межу максимальної ймовірності нетривіальної диференціальної характеристики, характерної для шифрів Маркова. "Калина" передбачає додавання ключів за модулем 232. Ця операція є нелінійною за GF(2) і може змінити кількість активних байт (для різниці, обчисленої за модулем 2). Більше того, ймовірність

такого перетворення залежить від вхідних даних, тому ймовірність всієї характеристики також залежить від вхідних даних. Таким чином, звичайні методи не можуть бути застосовані до "Калини". І це означає, що диференціальний (лінійний) криптоаналіз для «Калини» є практично неможливим.

Запропоновані вдосконалення шифру Rijndael, реалізовані в "Калині", дозволяють закрити виявлені потенційні вразливості в Rijndael. Це робить "Калину" стійкою до всіх відомих криптоаналітичних атак.

Цей висновок підтверджується результатами експертизи та досліджень, проведених в рамках національного конкурсу кандидатів на національний стандарт блочного симетричного шифрування.

1.4 Блоки підстановок

Таблиці підстановки або блок підстановки (або S-блоки – від Substitution box) є фундаментальним криптографічним компонентом, який відіграє важливу роль у реалізації властивості Шеннона про властивість заплутування в блокових шифрах. У двох основних стратегіях проектування блокових шифрів, мережі Фейстеля та мережі підстановки/перестановки, S-блоки утворюють єдину нелінійну частину блочного шифру. Стійкість шифру в значній мірі залежить від якості використовуваних S-блоки. S-блок приймає деяку кількість вхідних бітів, m, і перетворює їх у деяку кількість вихідних бітів, n, де n не обов'язково дорівнює m. S-блок m×n може бути у вигляді таблиці пошуку з 2^m слів по n бітів у кожному (див. рис. 1.2).

Задача пошуку оптимальних S-блоків є складною через те, що кількість перестановок, які відображають m біт в n біт, дуже велика навіть для малих значень m. Тому вичерпна перевірка всіх перестановок для знаходження хороших S-блоків для $m > 4$ є недоцільною [22]. На практиці S-блоки або проектуються, або генеруються випадковим чином. У кожному випадку важливо оцінити якість S-блоку.

Найважливішою властивістю S-блоку є стійкість до відомих атак (наприклад, диференціальний та лінійний криптоаналіз) та атак, які можуть бути винайдені в

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	4e	37	ea	8	e7	d5	a9	c8	6d	6c	8d	65	f4	56	7a	ae
10	42	89	0f	16	8c	e6	68	a1	0d	41	bf	b0	2d	99	54	bb
20	24	3a	62	79	e0	6	5c	32	0a	c2	49	91	ac	d3	95	e4
30	90	4f	14	db	60	2a	88	81	dc	46	22	de	b8	ee	5e	0b
40	b1	0	39	cf	53	fc	5b	d1	ed	6a	20	4a	be	cb	4c	58
50	b4	25	1f	8a	ba	a6	c6	78	2e	e8	1c	4b	74	dd	bd	8b
60	f6	b5	b9	9e	70	3	0e	3e	66	61	48	86	57	35	c1	1d
70	f7	93	f1	15	b7	3f	cc	fd	26	34	36	71	e5	a5	d8	31
80	5	23	e2	75	4	96	9a	c7	c3	7	18	eb	80	12	27	b2
90	44	13	3d	73	cd	97	17	0c	ec	c4	5f	64	7e	a7	5d	19
a0	8e	98	e9	df	e1	d9	94	f8	11	9b	69	ce	87	1e	55	28
b0	6f	77	2b	76	63	6b	c5	7c	7b	30	f2	fe	67	1	d7	ab
c0	5a	2c	b3	84	9	6e	a0	83	1a	52	1b	29	d6	3b	e3	2f
d0	33	aa	7f	a8	d0	4d	85	ef	fb	45	43	50	2	f9	3c	9f
e0	38	40	21	d2	51	9d	f5	a3	8f	bc	92	10	da	b6	ff	f3
f0	47	c9	af	c0	ca	59	f0	82	7d	ad	fa	9c	a2	d4	a4	72

Рисунок 1.2 – Приклад S-блоку у вигляді таблиці пошуку

майбутньому. Для оцінки криптографічної якості S-блоку було запропоновано ряд властивостей криптографічної якості S-блоку. Наприклад: нелінійність, збалансованість, критерій лавини, зворотність та несуперечність, диференціальна рівномірність, бітова незалежність [7]. На сьогоднішній день було запропоновано понад 20 параметрів, і через складність проблеми все ще існує можливість розробити нові заходи. Наприклад, інструмент з відкритим вихідним кодом для аналізу S-блоків "S-box, SET, Match" обчислює 17 параметрів. У деяких випадках критерії проектування та процес створення S-блоку залишаються нерозкритими.

1.5 Моделі та типи відомих атак на блочні шифри

Модель зловмисника визначає ресурси і можливості, якими він володіє, а також сценарій, за яким він атакує шифр.

Перше, що варто зазначити, це те, що в кожній моделі зловмисника передбачається, що він має доступ до дизайну шифру. Цей принцип був сформульований Керкхоффом у 1883 році і отримав назву принципу Керкхоффа.

Згідно з цим принципом, безпека криптосистеми не повинна залежати від секретності її дизайну. Вона повинна бути стійкою до атак, навіть якщо її дизайн став відомим зловмиснику.

Існує багато прикладів того, як криптографічні системи були зламані через те, що розробники не дотримувалися принципу Керкхоффа. Одним з відомих прикладів є Content Scramble System, яка використовувалася для захисту контенту DVD-Video і була повністю зламана після того, як її специфікація стала надбанням громадськості.

Крім принципу Керкхоффа, важливою частиною моделі зловмисника є сценарій атаки. Тут ми визначаємо тип даних, які зловмисник може отримати в свої руки, а також його потенційні знання про секретний ключ.

Дані, які зловмисник може використовувати у своїй атаці, це відкриті та зашифровані тексти. Залежно від способу збору цих даних розрізняють наступні сценарії атак:

– Атака тільки на шифрований текст. Зловмисник знає набір шифротекстів.

– Атака з відомим відкритим текстом. Зловмиснику відомі певні пари «відкритого-закритого»

– Атака за обраним відкритим текстом. Зловмисник може вибирати відкриті тексти і отримує їхні шифротексти.

– Атака за обраним шифрованим текстом. Зловмисник може вибирати шифротексти і отримує їхні відкриті тексти.

Найбільш небезпечними атаками є атаки за обраним відкритим текстом та за обраним шифрованим текстом. Це пов'язано з тим, що в цих атаках зловмисник має більший контроль над даними, які він може використовувати.

У загальному випадку, чим більше даних зловмисник має доступу, тим складніше розробити шифр, який буде стійким до його атак.

Лінійний та диференціальний криптоаналіз вважаються одними з найбільш важливих методів симетричного криптоаналізу [19], Відкриття диференціального криптоаналізу зазвичай приписують Елі Біаму та Аді Шаміру наприкінці 1980-х років. Вони опублікували численні атаки на різні блокові шифри та хеш-функції, включаючи теоретичні вразливості стандарту шифрування даних (DES). Біхем і Шамір зазначили, що DES напрочуд стійкий до диференціального криптоаналізу, але незначні зміни в алгоритмі можуть зробити його більш вразливим: 8-9 У 1994 році колишній член команди IBM DES Дон Копперсміт опублікував статтю. Про диференціальний криптоаналіз було відомо в IBM ще в 1974 році, і в ній зазначалося, що захист від диференціального криптоаналізу був метою проектування. DES був розроблений враховуючи стійкість до диференціального криптоаналізу, чого не було в інших сучасних шифрах. Першою мішенню для атак став блоковий шифр FEAL. Спочатку запропонована 4-раундова версія (FEAL-4) може бути зламана за допомогою лише 8 вибраних відкритих текстів, і навіть 31-раундова версія FEAL є вразливою до атак. На противагу цьому, дана схема успішно зламує DES приблизно в 247 обраних відкритих текстах.

Диференціальний криптоаналіз - це, як правило, атака на обраний відкритий текст. Це означає, що зловмисник повинен мати можливість отримати шифрований текст для обраного набору відкритих текстів або навіть напряму перебирати відкритий текст для шифрування. Однак, існують також розширення, які дозволяють проводити атаки з відомим відкритим текстом і тільки з зашифрованим текстом. Базовий метод використовує пари відкритих текстів, пов'язаних між собою постійною дельтою. Відмінності можуть бути визначені кількома способами, але найчастіше використовується операція виключного АБО (XOR). Потім зловмисник обчислює дельту відповідних зашифрованих текстів, намагаючись виявити статистичні закономірності в розподілі зашифрованих текстів. Отримана пара різниць називається різницею. Зловмисники аналізують дельти, оскільки їх статистичні властивості залежать від властивостей S-блоків, що використовуються для шифрування. Базова атака очікує, що одна конкретна відмінність у шифрованому тексті буде зустрічатися особливо часто. Таким чином

можна відрізнити шифр від випадкового. Більш складна варіація може відновити ключ швидше, ніж вичерпний перебір.

У одній з найпростіших форм відновлення ключа з використанням диференціального криптоаналізу зловмисник запитує шифротексти багатьох пар відкритих текстів і припускає, що дельти утримуються принаймні r - 1 раундів, де r - загальна кількість раундів. Потім зловмисник визначає, які раундові ключі (для останнього раунду) можливі, припускаючи, що різниця між блоками перед останнім раундом є фіксованою. Якщо ключ раунду короткий, цього можна досягти лише одним вичерпним розшифруванням пари зашифрованого тексту з використанням кожного можливого ключа раунду. Круглий ключ вважається правильним круглим ключем, якщо він розглядається як потенційний круглий ключ значно частіше, ніж будь-який інший ключ. Для деяких шифрів вхідна дельта повинна бути ретельно підібрана для успішної атаки. Проводиться аналіз внутрішньої структури алгоритму. Стандартна практика полягає в тому, щоб простежити ймовірні шляхи відмінностей на різних етапах шифрування, які називаються диференціальними властивостями. З тих пір, як диференціальний криптоаналіз став надбанням громадськості, він є основною проблемою розробників шифрів. Очікується, що нові розробки супроводжуватимуться доказами того, що алгоритми стійкі до цієї атаки, і багато алгоритмів, в тому числі Advanced Encryption Standard, довели свою стійкість до атак.

Диференціальний криптоаналіз, потужна криптоаналітична техніка, зробила революцію в аналізі криптографічних алгоритмів, зокрема, блокових шифрів, потокових шифрів і криптографічних хеш-функцій. Започаткований Елі Біамом та Аді Шаміром наприкінці 1980-х років, він відіграв ключову роль у виявленні вразливостей у численних криптографічних розробках.

В основі диференціального криптоаналізу лежить аналіз різниці між вхідними та вихідними даними в шифрі. Ретельно підбираючи вхідні пари (відкриті тексти), які відрізняються лише певними бітовими позиціями, а потім спостерігаючи за відповідними вихідними парами (зашифрованими текстами), аналітик намагається зрозуміти, як ці відмінності поширюються внутрішніми

циклами шифру. Таке ретельне дослідження дозволяє виявити закономірності та взаємозв'язки між різницями, виявляючи потенційні слабкі місця в структурі алгоритму.

Ці закономірності, відомі як диференціальні характеристики, є наріжним каменем диференціального криптоаналізу. Вони інкапсулюють очікувану поведінку різниць і являють собою потенційні вразливості, які можуть бути використані для відновлення ключа. Сила диференціальної характеристики безпосередньо пов'язана з її ймовірністю, яка вказує на те, як часто бажана поведінка проявляється в шифрі. Отже, статистичний аналіз великого масиву пар шифротекстів стає вирішальним для перевірки ефективності виявлених характеристик

Як тільки виявлено достатньо сильну диференціальну характеристику, її можна використовувати для запуску атак на відновлення ключа. Ці складні атаки передбачають маніпуляції з вхідними даними і спостереження за відповідними вихідними даними, поступово витягуючи інформацію про секретний ключ, який керує роботою шифру.

Диференціальний криптоаналіз має низку переваг, серед яких його широке застосування в різних криптографічних алгоритмах. Більше того, його ефективність була продемонстрована в численних випадках, виявляючи вразливості в реальних шифрах і спонукаючи до їх вдосконалення або заміни. Крім того, гнучкість методу дозволяє інтегрувати його з іншими методами криптоаналізу для підвищення ефективності атак.

Однак, важливо визнати, що диференціальний криптоаналіз має певні обмеження. Обчислювальна складність, пов'язана з аналізом складних шифрів і виявленням сильних характеристик, може створювати значні проблеми. Крім того, не всі шифри однаково піддаються цьому методу, деякі з них спеціально розроблені для протистояння таким атакам. Крім того, успіх деяких атак залежить від наявності великих обсягів даних, що не завжди можливо. Також, оскільки диференціальний криптоаналіз опирається на статистичний аналіз S-блоків

шифрів, то реалізації динамічних S-блоків на основі ключів роблять його абсолютно неефективним [16].

Незважаючи на ці обмеження, диференціальний криптоаналіз залишається наріжним каменем сучасного криптоаналізу. Його здатність використовувати витончені слабкі місця в криптографічних алгоритмах просунула вперед криптографічний дизайн і аналіз, формуючи ландшафт безпечного зв'язку та захисту інформації.

1.6 Постановка задачі

Блокові шифри є одними з найважливіших алгоритмів криптографії. Вони використовуються для захисту інформації в широкому спектрі застосувань, включаючи шифрування даних, електронний цифровий підпис і мережеву безпеку.

Одним з основних компонентів блокового шифру є S-блок. Він відіграє важливу роль у забезпеченні безпеки блокового шифру, оскільки він відповідає за перетворення вхідної інформації в непередбачуваний вихід.

Одним із методів криптоаналізу, який може бути використаний для атаки на блокові шифри, є диференціальний криптоаналіз. Диференціальний криптоаналіз передбачає дослідження того, як зміна вхідних даних впливає на вихідні дані. Якщо S-блок має низьку диференціальну рівномірність, то диференціальний криптоаналіз може бути використаний для знаходження ключів шифру.

Дослідження стійкості блокових шифрів включають вивчення стійкості популярних шифрів та їх мініверсій [4], а також розробку нових методів проектування криптографічно стійких S-блоків [10]. Деякі з цих методів досліджують вплив на криптографічні характеристики блоків [8], але не було досліджень, які б безпосередньо досліджували ступінь стійкості S-блоку до диференціального аналізу з різними рівнями характеристики диференціальної рівномірності.

Тому метою цієї кваліфікаційної роботи є дослідження залежності стійкості S-блоків до диференціального криптоаналізу від їх диференціальної рівномірності. Для досягнення цієї мети необхідно вирішити такі задачі:

– Розробити алгоритм генерації S-блоків з заданим ступенем диференціальної рівномірності.

– Розробити алгоритм диференціального криптоаналізу S-блоків в межах одноразундового блокового шифру.

– Провести диференціальний криптоаналіз S-блоків з різними ступенями диференціальної рівномірності.

Результати цієї роботи будуть корисними для розробників блокових шифрів, оскільки вони дозволять їм краще зрозуміти, як диференціальна рівномірність S-блоків впливає на їхню стійкість до диференціального криптоаналізу.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ДЛЯ ГЕНЕРАЦІЇ S-БЛОКІВ ТА ДИФЕРЕНЦІАЛЬНОГО КРИПТОАНАЛІЗУ

2.1 Методи генерації S-блоків

Сучасні блокові шифри часто є ітераціями з декількох раундів. Кожен раунд(який повинен залежати від ключа) складається з шару плутанини і шару дифузії. Блокові шифри можуть бути побудовані використовуючи добре відому структуру, таку як мережа Фейстеля, мережа підстановок і перестановок (SPN) або структура Лай-Мессі.

Криптографічні властивості S-блоків пов'язані з застосуванням декількох логічних атак на шифри, а саме лінійної атаки, диференціальної атаки, та алгебраїчних атак, застосування яких для більшості шифрів має більш теоретичне значення і не є повністю дослідженим [1] , але представляє певну загрозу і про неї слід пам'ятати розробникам наступного покоління блокових шифрів. З цієї причини S-блок повинен задовольняти різним критеріям для забезпечення високого рівня захисту від таких атак.

Окрім лінійних, диференціальних та алгебраїчних атак, на сьогоднішній день найбільші атаки на криптографічні алгоритми базуються на спостереженні за фізичними процесами в криптографічному пристрої. В літературі цей вид атак отримав назву атаки побічних каналів (side-channel attacks, SCA) [9]. Прикладами таких атак є простий аналіз потужності (SPA), диференціальний аналіз потужності (DPA). Подібні атаки становлять видиму загрозу для блочних шифрів з блоками підстановки [13].

S-блоки є одним з найважливіших елементів блокових шифрів, оскільки вони відповідають за нелінійність шифрування, яка є необхідною для підвищення його стійкості до криптоаналізу.

Існує чотири основних класифікації методів генерації S-блоків:

– Алгебраїчні побудови засновані на використанні математичних об'єктів, таких як групи, поля та кільця. Ці методи дозволяють отримати S-блоки з високими

криптографічними властивостями, але вони є складними і вимагають значних знань і досвіду.

– Псевдовипадкова генерація заснована на використанні псевдовипадкових послідовностей. Ці методи дозволяють отримати S-блоки з високою швидкодією і простотою реалізації, але вони можуть призвести до S-блоків з меншими криптографічними властивостями.

– Евристичні методи засновані на використанні експертних знань і досвіду. Ці методи дозволяють отримати S-блоки з прийнятними криптографічними властивостями, але вони не гарантують високу стійкість до криптоаналізу.

– Будови від малих до великих S-блоків засновані на використанні S-блоків меншого розміру. Ці методи дозволяють отримати S-блоки з високими криптографічними властивостями, але вони можуть бути трудомісткими.

Алгебраїчні побудови S-блоків є одним з найбільш поширених методів. Вони використовуються в багатьох відомих блокових шифрах, таких як DES, AES, IDEA та Blowfish.

Існує кілька підходів до алгебраїчного проектування S-блоків. Один з підходів полягає в тому, щоб використовувати теорію груп для забезпечення того, щоб S-блок був аффінним перетворенням.

Інший підхід полягає в тому, щоб використовувати теорію полів для забезпечення того, щоб S-блок був лінійним перетворенням.

Псевдовипадкова генерація S-блоків є більш новим методом, який стає все більш популярним. Він дозволяє отримати S-блоки з високою швидкодією і простотою реалізації. Існує кілька алгоритмів для псевдовипадкової генерації S-блоків. Один з таких алгоритмів полягає в тому, щоб використовувати метод генерації псевдовипадкових послідовностей на основі криптографічних хеш-функцій. Подібною методикою керувалися розробники шифру «Калина» [3].

Інший алгоритм полягає в тому, щоб використовувати метод генерації псевдовипадкових послідовностей на основі криптографічних потокових шифрів.

Евристичні методи генерації S-блоків є найбільш простими і швидкими методами, як наприклад описано у [12]. Вони дозволяють отримати S-блоки з

прийнятними криптографічними властивостями, але вони не гарантують високу стійкість до криптоаналізу.

Існує кілька підходів до евристичного проектування S-блоків. Один з підходів полягає в тому, щоб використовувати таблиці псевдовипадкових чисел або таблиці чисел на основі відбитків пальців [21].

Інший підхід полягає в тому, щоб використовувати таблиці, які були отримані шляхом ручного проектування.

Будови від малих до великих S-блоків є відносно новим методом, який стає все більш популярним. Вони дозволяють отримати S-блоки з високими криптографічними властивостями, як наприклад [15].

Ці методи засновані на використанні S-блоків меншого розміру. Спочатку будуються S-блоки меншого розміру, а потім вони комбінуються для отримання S-блоків більшого розміру.

Описані алгоритми дають хороші результати для побудови бієктивних S-блоків лише за одним з основних критеріїв, але завдання стає набагато складнішим, коли необхідно враховувати більше властивостей одночасно [18]. Зазвичай розробники розбивають всі властивості, які необхідно врахувати, на декілька частин, щоб шукати S-блок поетапно. Наприклад, спочатку шукають набір S-блоків з оптимальною диференціальною однорідністю та лінійністю. Потім вони вичерпують S-блоки з цієї множини, щоб задовольнити інші вимоги, такі як відсутність фіксованих точок, високе число розгалужень і так далі.

В межах поточного дослідження S-блоків ми будемо використовувати для роботи метод псевдовипадкової генерації. Оскільки ми будемо працювати з невеликими за розмірами блоками, то генерація великої кількості блоків для відбору необхідних не стане проблемою.

2.2 Опис алгоритму диференціального криптоаналізу

Диференціальний криптоаналіз - це метод криптоаналізу, який використовує диференціальні властивості шифру для відновлення секретного ключа.

Загальні кроки, необхідні для проведення диференціального криптоаналізу, наступні:

– Побудова таблиці диференціального розподілу.

– Вибір диференціала з високим показником диференціальної рівномірності.

– Пошук пар відкритих повідомлень з обраним диференціалом.

– Пошук ключа.

Для того, щоб здійснити диференціальну атаку на блоковий шифр, зловмисник шукає всі можливі витягнуті особливості як з відкритого, так і з зашифрованого текстів. Ці особливості відомі як диференціальні характеристиками, які описують відношення між відкритим і зашифрованим текстом (див. рис. 2.1).

Відмінності між входами і виходами будь-якої заданої структури S-блоку представляють собою відношення між набором вхідних бітів і набором вихідних бітів після кожної заміни. Зловмисники досліджують диференціал для кожного входу S-блоку.

Виходячи з цього, отримані диференціали можна звести в таблицю, стовпчики якої представляють диференціали між вхідними парами, а рядки – диференціали між парами на виході. Отримана таблиця відома як таблиця диференціального розподілу. Таким чином, вона містить повну характеристику, що дозволяє обчислити ймовірність появи інформативних відкритих і зашифрованих текстів, які можуть призвести до отримання ключа шифрування.

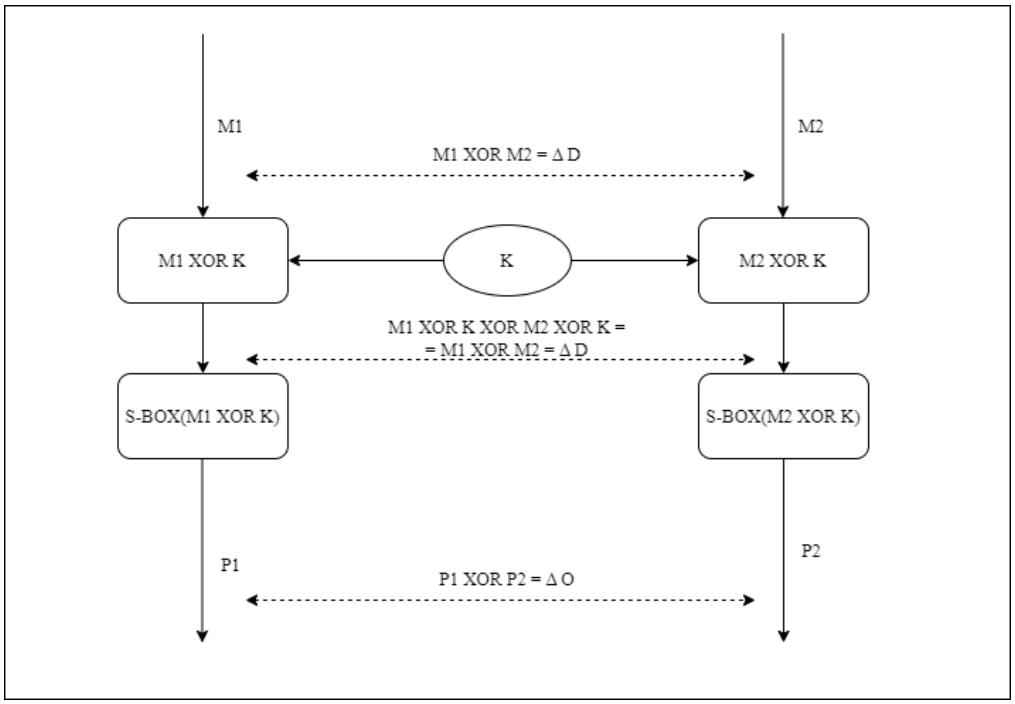

Рисунок 2.1 – Схема пошуку диференціалу для відкритого і закритого повідомлення

Таким чином, таблиця диференціального розподілу використовується або для відновлення або для опису здатності блочного шифру протистояти диференціальним атакам, намагаючись зменшити значення ймовірності зменшити значення ймовірності [11], які наведені в таблиці. Приклад таблиці диференціального розподілу наведено на рисунку 2.1.

Диференціальна рівномірність в свою чергу визначається, як максимальне значення з таблиці диференціального розподілу (окрім значення на перехресті диференціалів-нулів в лівому верхньому кутку).

Оскільки в межах роботи ми намагаємося досліджувати стійкість самих S-блоків до диференціального криптоаналізу, то найкращим рішенням буде використовувати шифр з мінімальною кількістю інших шарів та операцій, які можуть впливати на результати аналізу.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	i/o
32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
-	2	-	-	4	-	2	4	2	2	-	-	2	-	2	-	1
-	-	-	2	2	-	2	2	2	-	2	-	4	-	-	4	2
-	2	2	-	-	-	2	2	4	-	-	2	-	2	-	-	3
-	-	2	6	-	-	-	-	4	-	-	-	-	-	2	2	4
-	-	2	-	2	2	-	-	-	-	-	2	-	-	2	2	5
-	-	-	-	4	-	-	2	4	2	-	-	4	-	-	-	6
-	2	-	4	-	-	-	-	-	-	-	-	-	-	-	2	7
-	-	2	-	-	4	6	-	-	2	2	-	-	-	4	-	8
-	6	2	-	2	-	-	2	-	-	2	-	-	-	-	2	9
-	2	2	-	-	-	6	-	-	4	2	-	4	-	-	-	10
-	-	-	-	-	-	2	4	2	-	-	-	-	2	2	-	11
-	-	-	-	4	2	2	-	-	-	4	-	-	2	2	-	12
-	-	-	-	-	-	-	-	-	2	2	-	2	-	2	-	13
-	2	4	-	-	2	-	-	-	2	2	-	-	-	4	-	14
-	-	-	-	2	2	2	-	2	2	-	-	-	2	-	-	15

Рисунок 2.1 – Таблиця диференціального розподілу

Структура експериментального одноразундового шифру, який буде використовуватися для експериментального дослідження, зображена на рисунку 2.2.

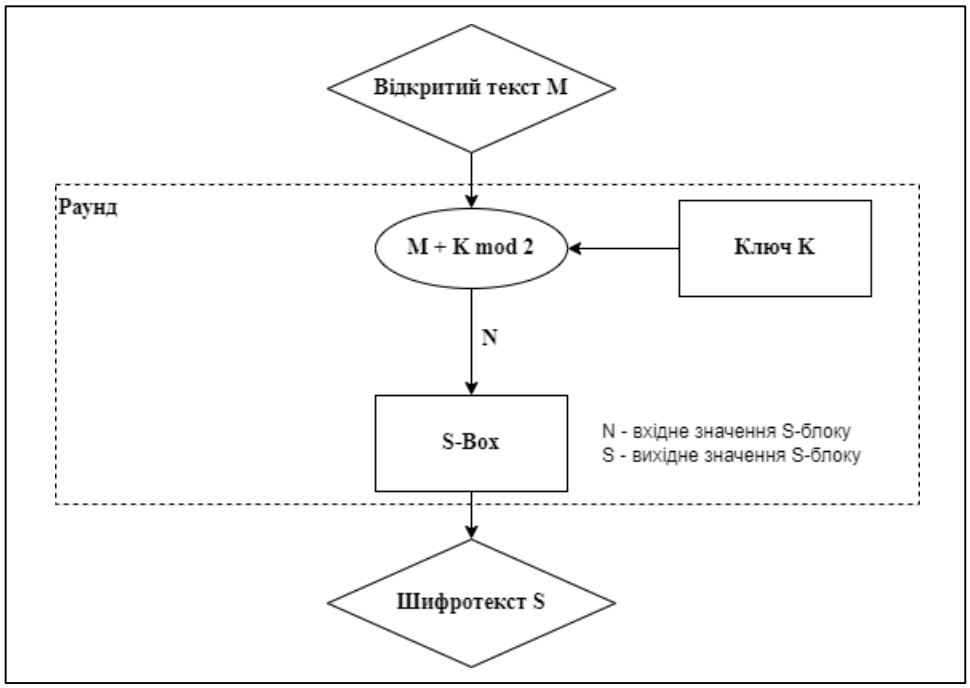

Рисунок 2.2 – Експериментальний одноразундовий шифр

Ескпериментальний шифр буде складатися лише з одного раунду, який в свою чергу реалізовуватиме лише дві операції – додавання ключа та підстановку за допомогою S-блока.

2.3 Обгрунтування вибору мови програмування та середовища розробки програмного забезпечення

Для написання програмного забезпечення, яке буде реалізувати згадані алгоритми, потрібно обрати мову програмування. Одним з найкращих варіантів буде саме Python, адже Python - це сучасна, проста у вивченні, об'єктно-орієнтована мова програмування. Вона має потужний набір вбудованих типів даних типів даних і простих у використанні керуючих конструкцій. Оскільки Python є інтерпретованою мовою, її найпростіше вивчати просто переглядаючи та описуючи інтерактивні сесії. Вона використовується у величезній кількості додатків завдяки різноманітним завдяки різноманітним стандартним бібліотекам, які постачаються разом з нею, а також її здатності інтегруватися з іншими мовами та використовувати їхні можливості. Python можна використовувати для написання скриптів, веб-скрепінгу та створення наборів даних. Python – це також мова веб-програмування, тому вона взаємодіє з інтернетом. Вона вміє отримувати та надсилати веб-запити та спілкуватися з базами даних.

Ось її основні переваги:

– На відміну від C, C++, C# та Java, синтаксис мови Python дуже простий і зручний для розробників, тобто базовий код можна легко вивчити і написати за пару годин або днів. Python - одна з найпростіших для вивчення мов.

– Будучи технологією з відкритим вихідним кодом, Python має велику спільноту кодерів, спеціалістів з обробки даних тощо, які роблять свій внесок у процес розробки Python, а також великий набір бібліотек для різних цілей. Існує безліч онлайн-форумів, де люди обговорюють і діляться знаннями про Python з іншими програмістами, а також ведуть бесіди на різні теми, пов'язані з мовою.

Користувачі Windows, Mac або Linux можуть завантажити останню версію Python безкоштовно.

– Існує велика кількість фреймворків графічного інтерфейсу, доступних у Python. Більшість програмного забезпечення сьогодні поставляється з добре розробленим зручним графічним інтерфейсом користувача, щоб зробити результати більш наочними.

– Python сумісна з широким спектром апаратних платформ. Код на Python може виконуватися на Linux, MacOs, Microsoft Windows тощо. Кодер може безперешкодно запускати один і той самий код на різних платформах до тих пір, поки не буде включено строго системно-залежного коду.

– Це мова високого рівня. Оскільки Python є мовою програмування високого рівня, кодеру не потрібно приділяти багато уваги архітектурі апаратного забезпечення під час написання коду.

– Типи даних для змінних у Python не визначаються під час написання коду, натомість Python автоматично визначає тип і відповідно розподіляє пам'ять, тим самим захищаючи користувача від помилок невідповідності типів даних.

– Python постачається з багатими динамічними структурами даних, такими як списки, множини, кортежі, словники тощо, як частина стандартної інсталяції, тому він миттєво готовий до використання.

– Код на Python виконується рядок за рядком без необхідності попередньої компіляції, що робить налагодження легшим та ефективнішим для початківців порівняно з іншими мовами, що компілюються.

– Однією з чудових особливостей Python є універсальність, тобто вона підтримує виконання коду, написаного на мовах C, C++, Java тощо. Користувач може викликати бібліотеки та функції таких мов як C, C++ тощо з Python за допомогою окремих бібліотек.

– Python можна вважати гібридною мовою, яка може підтримувати як повністю об'єктно-орієнтовані концепції програмування, так і структурований стиль програмування. Фактично, все в Python є об'єктом. ООП в Python допомагає

визначати об'єкти реального світу в програмуванні і описує, які відносини існують між об'єктами.

Саме завдяки цим перевагам, Python була обрана мовою програмування для написання коду для цього дослідження. Наступним кроком є вибір середовища розробки. І PyCharm, як комплексне інтегроване середовище розробки (IDE), доповнює сильні сторони Python. Інтелектуальне завершення коду та підказки підвищують продуктивність і точність, зменшуючи кількість помилок і пришвидшуючи розробку. Інтегровані інструменти налагодження та тестування дозволяють ефективно виявляти та вирішувати проблеми, забезпечуючи якість та надійність коду. PyCharm спрощує організацію проекту, навігацію та рефакторинг, полегшуючи управління складними дослідницькими кодовими базами. Крім того, безшовна інтеграція PyCharm з науковими бібліотеками Python надає спеціалізовані функції для аналізу та візуалізації даних, спрощуючи робочий процес дослідження.

Особливі переваги Python та PyCharm для дослідження S-блоків включають підтримку побітових операцій та гнучких структур даних, які ідеально підходять для представлення та маніпулювання S-блоками та їхніми властивостями.

Насамкінець, зручність Python, великі наукові бібліотеки та можливості швидкого створення прототипів у поєднанні з інтелектуальними функціями розробки PyCharm роблять їх потужним та ефективним вибором для проведення досліджень S-блоків у блокових шифрах. Зручність і доступність цих інструментів сприяють співпраці та відтворюваності результатів у дослідницькій спільноті.

2.4 Розробка програмного забезпечення для генерації S-блоків

Для реалізації алгоритму псевдовипадкової генерації S-блоків, нам потрібно буде використовувати стандартну бібліотеку мови Python random. В лістингу 2.1 наведено код функції, яка генерує псевдовипадковий S-блок. Для цього в функції запускається цикл від 0 до максимального значення при вказаній кількості бітів. Всередині цього циклу для кожного значення і з множини усіх можливих значень

вибирається випадкове значення і записується в словник. Таким чином поступово після закінчення циклу словник буде заповнений і буде новим S-блоком.

Лістинг 2.1 – Функція псевдовипадкової генерації S-блока

```
def generate_sbox(bits=BITS, console_print=False):
    max_value = 2 ** bits
    bits_list = [i for i in range(max_value)]

    s_box = {}

    for i in range(max_value):
        ind = randrange(0, len(bits_list))
        s_box[i] = bits_list.pop(ind)

    if console_print:
        for key, value in s_box.items():
            print(f"{key}: {value}, ", end="")

    return s_box
```

В лістингу 2.2 наведена функція, яка дозволяє визначити диференціальну рівномірність переданого їй S-блока. Для цього функція викликає функцію знаходження таблиці диференціального розподілу, яка буде розглянута в наступних підрозділах. З цієї таблиці нам необхідно знайти наступне після максимального значення. Це і буде диференціальна рівномірність нашого S-блока.

Лістинг 2.2 – Функція перевірки диференціальної рівномірності

```
def check_diff_uniformity(s_box, bits=BITS):
    (ddt, ddt_input_pairs) =
differential_table.generate_dd_table(s_box, bits)
    values = []
    for i in range(len(ddt)):
```

Продовження лістингу 2.2

```
    for j in range(len(ddt[i])):
            if ddt[i][j] != "-":
                values.append(ddt[i][j])
    values.remove(max(values))
    return max(values), ddt, ddt_input_pairs
```

2.5 Реалізація програмного забезпечення для диференціального криптоаналізу

В лістингу 2.3 наведено код функції, яка дозволяє згенерувати таблицю диференціального розподілу для S-блока. Для цього запускається два цикли, один всередині іншого, які проходяться по всіх можливих вхідних значеннях шифру. Для кожної пари значень проводиться операція XOR для визначення диференціальної різниці конкретної пари. Потім ці пари проводять через S-блок і обчислюють диференціальну різницю для пари на виході. Результати вносять в таблицю, яка є двовимірним масивом. На перехресті значень вхідний-вихідний диференціал в таблиці додають 1. А конкретні значення вхідних пар, які дали ці диференціали заносять в окремий масив, який буде використовуватися в подальшому для аналізу.

Лістинг 2.3 – Функція генерації таблиці диференціального розподілу

```
    def generate_dd_table(s_box=S_BOX, bits=BITS,
console_print=False):
        """Generate DDT and all the input pairs that DDT consist of
in form of two two-dimensional lists."""
        max_value = 2 ** bits
         # set up empty table
        table = [["-" for j in range(max_value)] for i in
range(max_value)]
        table_of_pairs = [[[] for j in range(max_value)] for i in
range(max_value)]
```

Продовження лістингу 2.3

```
    # check XOR difference for every possible pair of input
        for i in range(max_value):
            for j in range(max_value):
                input_dif = i ^ j
                output_diff = s_box[i] ^ s_box[j]
                if table[output_diff][input_dif] == "-":
                    table[output_diff][input_dif] = 0
                table[output_diff][input_dif] += 1
                table_of_pairs[output_diff][input_dif].append((i,
j))
                print(f"Progress: {(i*max_value+j+1)/(max_value *
max_value)*100}%")

        if console_print:
            print_table(table, [i for i in range(max_value)], [i
for i in range(max_value)], bits)

        return table, table_of_pairs
```

В лістингу 2.4 наведено код функції, яка використовується для шифрування відкритого тексту за допомогою реалізації простого експериментального шифру, структура якого була згадана в розділі 2.3. В цій функції спочатку виконується операція XOR між відкритим текстом та ключем, після чого отримане значення пропускається через S-блок. Отриманий результат і є зашифрованим текстом.

Лістинг 2.4 – Функція шифрування відкритого тексту

```
    def encrypt(s_box=S_BOX, bits=BITS, key1=KEY1,
message=MESSAGE):
        # 1. xor message with first part of the key
        secret = message ^ key1
        # 2. push result through s-box
        secret = s_box[secret]
```

Продовдження лістингу 2.4

```
        # pretty print results
        print(f"Open: {message:0{bits}b} ({message})\nKey:
{key1:0{bits}b} ({key1})\n"
              f"Secret: {secret:0{bits}b} ({secret})")

        return secret
```

Наведений в лістингу 2.4 код використовується як для шифрування повідомлень, так і для того, щоб шукати пари відкритих текстів, які необхідні для виконання диференціального криптоаналізу. Повні версії використаного в роботі коду наведено в додатку Б.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СТІЙКОСТІ S-БЛОКІВ

3.1 Методика проведення дослідження

Дослідження буде проводитися за допомогою експерименту. На початку ми оберемо 5 різних значень диференціальної рівномірності. Для кожного з значень диференціальної рівномірності буде згенеровано 10 S-блоків. Кожен згенерований S-блок буде застосований для шифрування 10 сетів повідомлення-ключ на основі однораундового шифру і для кожного сету буде проведено диференціальний криптоаналіз. При кожному диференціальному криптоаналізі вимірюється кількість затрачених на процес пошуку пар вхідних повідомлень і самих ключів циклів. В межах шифру будемо оперувати блоками по 32 біти.

Для експерименту можна використовувати будь-який метод генерації S-блоків, який дозволяє отримати S-блоки з заданим значенням диференціальної рівномірності. В цьому експерименті ми будемо використовувати метод псевдовипадкової генерації S-блоків, адже він дозволяє досить швидко і ефективно генерувати велику кількість малих за розміром S-блоків з необхідними властивостями.

Основною вимірюваною величиною є кількість затраченого на диференціальний криптоаналіз часу. Вимірювання буде проводитися за рахунок логування часу в коді розробленого програмного забезпечення.

Експеримент буде проводитися в наступному порядку:

– Генерація 10 S-блоків для кожного з 5 значень диференціальної рівномірності методом випадкової генерації.

– Для кожного S-блоку проводить шифрування 10 повідомлень (з різними ключами).

– Для кожного зашифрованого повідомлення проводиться диференціальний криптоаналіз.

– Вимірюється кількість затраченого на кожний диференціальний криптоаналіз часу.

Для обробки результатів експерименту будуть використані наступні методи:

– Середнє значення кількості затрачених циклів.

– Візуалізація отриманих результатів за допомогою графіків.

На основі результатів експерименту будуть зроблені висновки про вплив диференціальної рівномірності S-блоків на ефективність диференціального криптоаналізу.

3.2 Аналіз результатів експериментальних досліджень

Для проведення експерименту було обрано 5 різних характеристик диференціальної рівномірності: 6/32, 8/32, 10/32, 16/32, 18/32. Хоча теоретично S-блоки повинні бути найбільш стійкими при мінімальній диференціальній рівномірності, наприклад 2/32 чи 4/32, проте згенерувати такі блоки методом псевдовипадкової генерації досить важко і затратно. Тому ми обмежимося S-блоками з дещо вищою диференціальною рівномірністю.

Для кожного обраного значення характеристики було згенеровано по 10 S-блоків. Створення кожного блоку виконувалося методом псевдовипадкової генерації. Для цього використовувався псевдовипадковий генератор стандартної бібліотеки Python.

Після генерації S-блоків було проведено 10 операцій шифрування і диференціального криптоаналізу для кожного з них. Для кожної операції шифрування використовувався один із 20 S-блоків, що були згенеровані для відповідного значення характеристики диференціальної рівномірності.

На рисунках 3.1 та 3.2 зображено результати двох операцій для S-блоків з різною диференціальною рівномірністю у вигляді таблиці.

Усереднені результати для кожної операції S-блоків продемонстровані на рисунках 3.3-3.7. На цих рисунках зображено середню затрачену кількість циклів для атаки на кожен згенерований S-блок.

Ключ K1, Повідомлення M1

Диференціальна рівномірність	Блок 1	Блок 2	Блок 3	Блок 4	Блок 5	Блок 6	Блок 7	Блок 8	Блок 9	Блок 10	Середні
6/32	0.1760	0.2750	0.0630	0.3000	0.1900	0.1300	0.0100	0.3250	0.0660	0.3250	0.19
8/32	0.0360	0.2090	0.2080	0.1980	0.2040	0.0840	0.0900	0.1650	0.0450	0.0810	0.13
10/32	0.2530	0.0600	0.1300	0.1430	0.0900	0.2200	0.0480	0.0600	0.0300	0.3220	0.14
16/32	0.0840	0.0330	0.0840	0.2240	0.1700	0.0120	0.1960	0.1170	0.1820	0.0630	0.12
18/32	0.1430	0.0810	0.1950	0.0540	0.0720	0.1400	0.0260	0.2100	0.0880	0.1430	0.12

Рисунок 3.1 – Результати проведення операції з K1, M1 для різних блоків

Ключ K2, Повідомлення M2

Диференціальна рівномірність	Блок 1	Блок 2	Блок 3	Блок 4	Блок 5	Блок 6	Блок 7	Блок 8	Блок 9	Блок 10	Середні
6/32	0.2080	0.0360	0.1530	0.0810	0.2610	0.3500	0.0770	0.1820	0.1500	0.2900	0.18
8/32	0.0110	0.0840	0.1200	0.0120	0.1690	0.2100	0.1080	0.0780	0.2250	0.0390	0.11
10/32	0.0390	0.1680	0.0140	0.1120	0.0650	0.0630	0.0330	0.0630	0.0420	0.2210	0.08
16/32	0.2210	0.1320	0.0240	0.1800	0.1950	0.0200	0.0390	0.2160	0.1690	0.0440	0.12
18/32	0.1430	0.1260	0.0840	0.0140	0.0900	0.1560	0.0540	0.0140	0.0180	0.0100	0.07

Рисунок 3.2 – Результати проведення операції K2, M2 для різних блоків

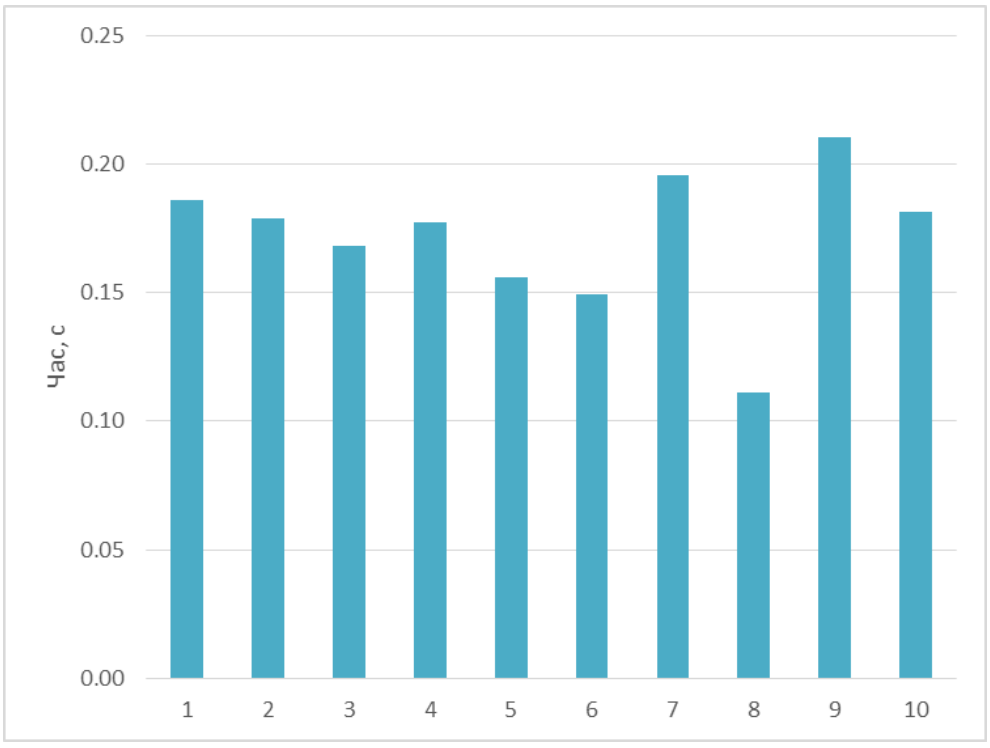

Рисунок 3.3 – Середні значення затраченого часу для S-блоків з характеристикою 6/32

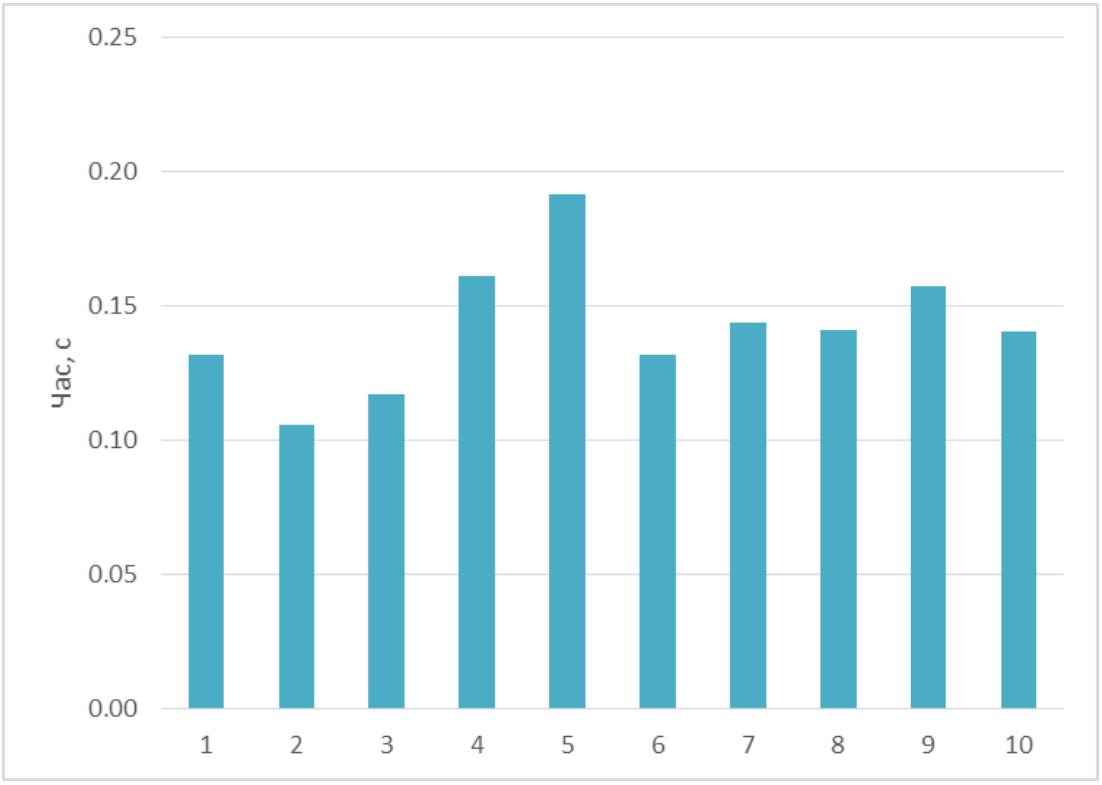

Рисунок 3.4 – Середні значення затраченого часу для S-блоків з характеристикою 8/32.

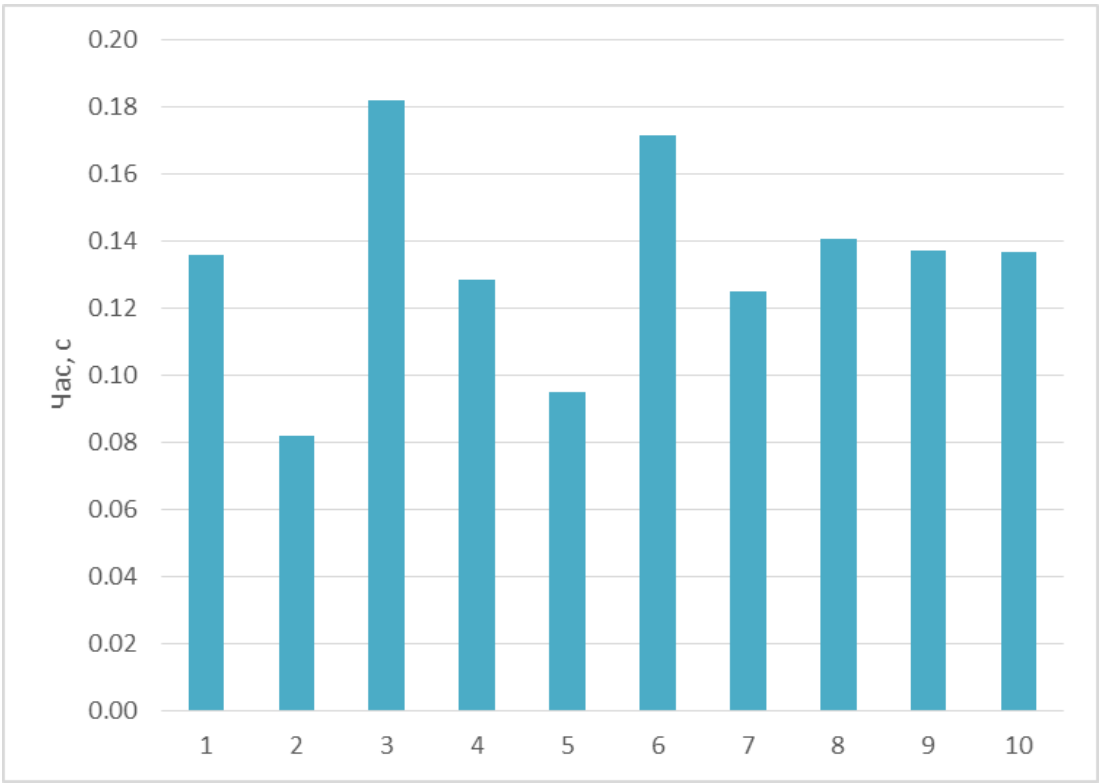

Рисунок 3.5 – Середні значення затраченого часу для S-блоків з характеристикою 10/32.

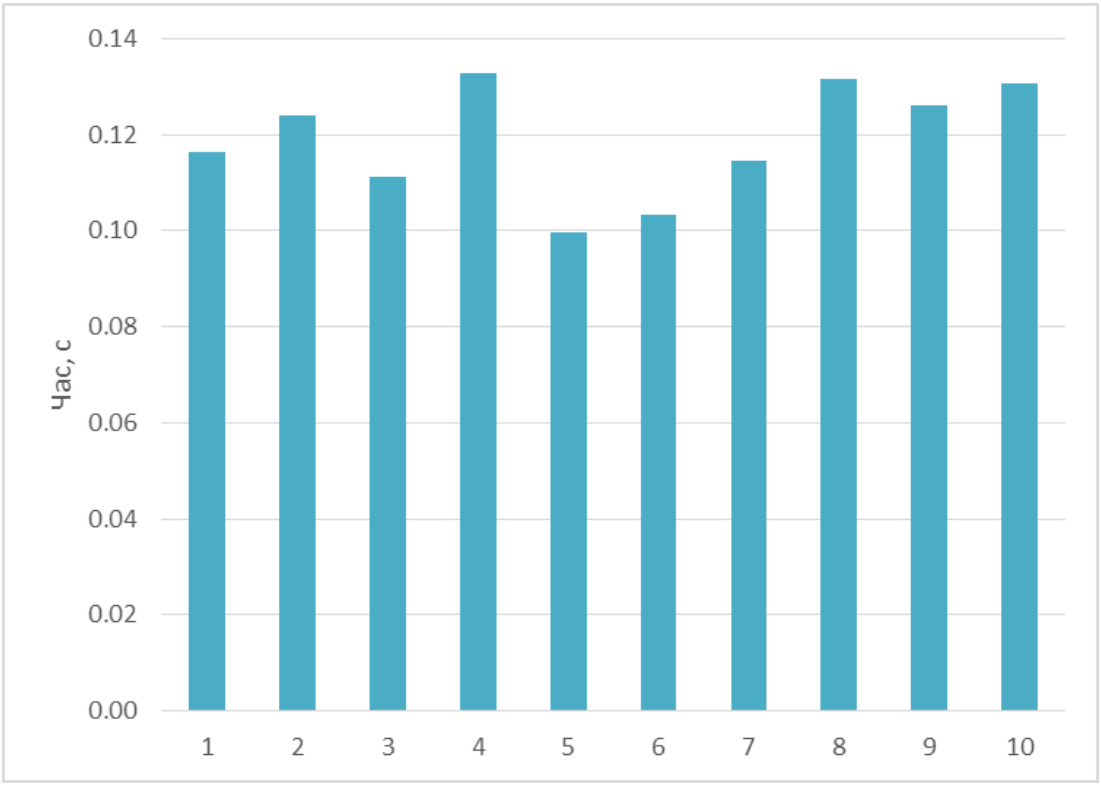

Рисунок 3.6 – Середні значення затраченого часу для S-блоків з характеристикою 16/32

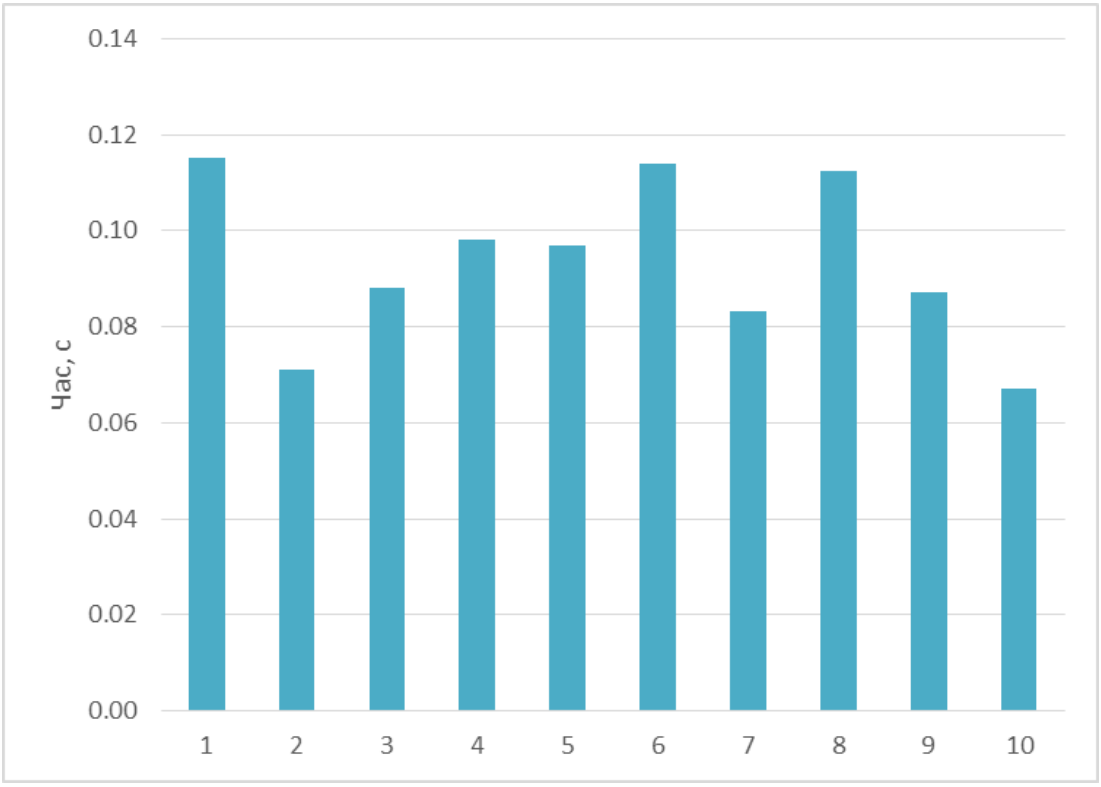

Рисунок 3.7 – Середні значення затраченого часу для S-блоків з характеристикою 18/32

На поданих графіках чітко прослідковується залежність кількості затраченого на аналіз часу від диференціальної характеристики: чим більша характеристика, тим менше часу затрачається. Це означає, що S-блоки з більш рівномірною диференціальною характеристикою є більш стійкими до диференціального криптоаналізу.

На рисунку 3.8 подані усереднені значення затраченого часу для кожної групи S-блоків. З графіку можна зробити висновок, що стійкість S-блоків з диференціальною рівномірністю близько 8/32 чи 10/32 має відносно меншу різницю в затраченому часі, ніж блоки з іншими значеннями. Це означає, що S-блоки з цими значеннями диференціальної характеристики мають близьку стійкість до диференціального криптоаналізу.

Цей висновок має важливе значення для розробників шифрів. Він вказує на те, що при виборі S-блоків варто обирати або суттєво кращу характеристику, ніж 10/32, або обирати 10/32 і не затрачати ресурси на пошук блоку з рівномірністю 8/32, стійкість якого хоч і буде відносно більшою, але не суттєво.

На основі проведених експериментальних досліджень можна зробити такі висновки щодо залежності стійкості S-блоків до диференціального криптоаналізу від диференціальної рівномірності: висока диференціальна рівномірність є важливою характеристикою для забезпечення стійкості S-блоків до диференціального криптоаналізу. Однак, диференціальна рівномірність не є єдиною характеристикою, яка визначає стійкість S-блоків до диференціального криптоаналізу. Існують інші фактори, такі як структура S-блоків та їх взаємодія з іншими компонентами шифру, які також можуть впливати на стійкість до диференціального криптоаналізу.

Для подальшого розвитку дослідження залежності стійкості S-блоків до диференціального криптоаналізу від диференціальної рівномірності можна рекомендувати такі напрями досліджень:

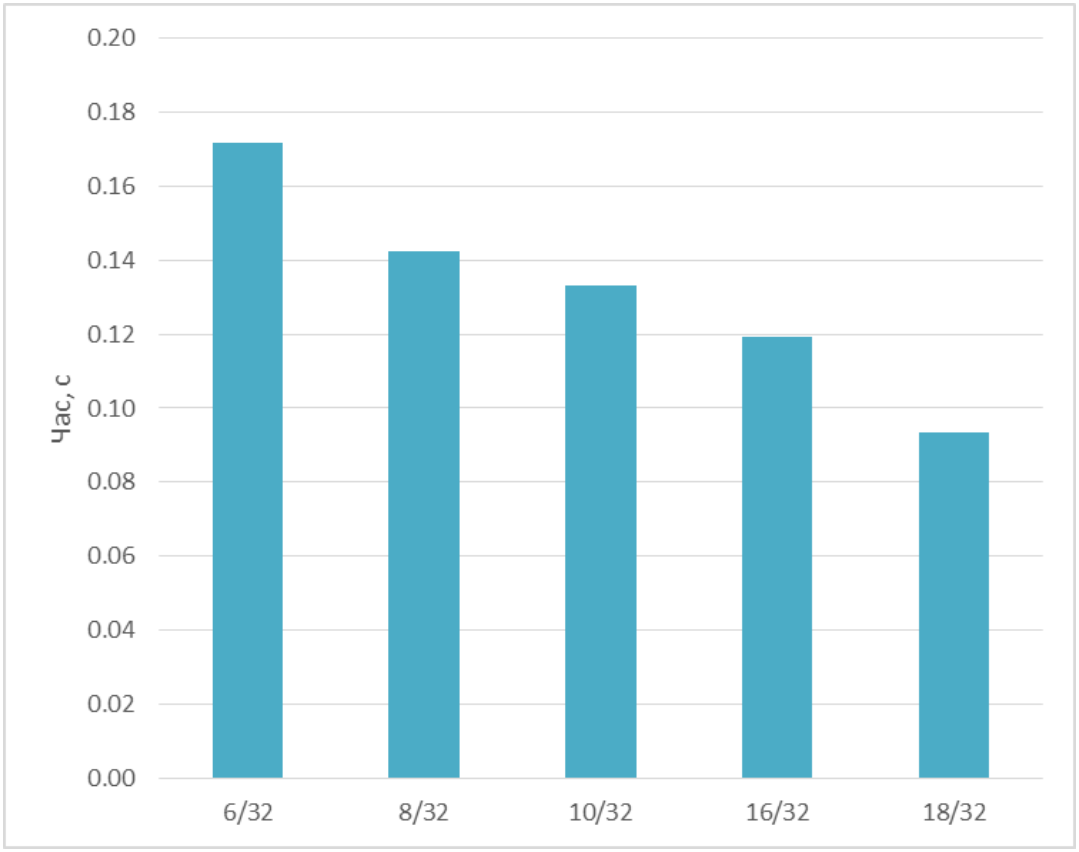

Рисунок 3.8 – Середні значення затраченого часу для S-блоків з різними характеристиками

– Розширення досліджень на більшу кількість S-блоків різної структури. Це дозволить отримати більш репрезентативне уявлення про те, як диференціальна рівномірність впливає на стійкість S-блоків до диференціального криптоаналізу.

– Врахування впливу структури S-блоків та їх взаємодії з іншими компонентами шифру на стійкість до диференціального криптоаналізу. Це дозволить розробити більш точні методи оцінки стійкості S-блоків до диференціального криптоаналізу.

Ось деякі конкретні експериментальні дослідження, які можна провести в рамках цих напрямів:

– Вивчити вплив різних структур S-блоків на їх стійкість до диференціального криптоаналізу. Це дозволить розробити більш точні методи оцінки стійкості S-блоків до диференціального криптоаналізу.

– Вивчити вплив взаємодії S-блоків з іншими компонентами шифру на їх стійкість до диференціального криптоаналізу. Це дозволить розробникам шифрів створювати S-блоки з підвищеною стійкістю до диференціального криптоаналізу.

Виконання цих досліджень дозволить краще зрозуміти, як диференціальна рівномірність впливає на стійкість S-блоків до диференціального криптоаналізу. Це, в свою чергу, призведе до розробки більш стійких до диференціального криптоаналізу S-блоків і, відповідно, більш стійких до диференціального криптоаналізу шифрів.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Темою моєї магістерської роботи є дослідження стійкості S-блоків до диференціального криптоаналізу. Оскільки S-блоки завжди є частиною криптографічних систем, що використовують комп'ютерне обладнання, включаючи комп'ютери та периферійні пристрої, важливо дотримуватися вимог охорони праці та техніки безпеки.

Охорона праці, по суті, пов'язана з турботою про людей у процесі використання праці і розглядається як захист працездатності людини. Охорона праці є одним із пріоритетних напрямів соціальної політики України. З цією метою Конституція України визначає право кожного на належні, безпечні і здорові умови праці як одне з основних соціальних прав громадян і забороняє використання праці жінок і неповнолітніх на шкідливих для їхнього здоров'я роботах. Право на охорону здоров'я також закріплено в Основах законодавства України про охорону здоров'я.

Що стосується комп'ютерних систем, то існують правила охорони праці, яких необхідно дотримуватися при роботі з криптографічними системами.

Умови та організація роботи з усіма типами візуальних дисплейних терміналів вітчизняного та зарубіжного виробництва на електронно-променевих трубках, що використовуються в електронно-обчислювальних машинах загального користування та персональних комп'ютерах, визначаються "Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин" (ДСанПіН 3.3.2.007-98), затвердженими постановою Головного державного санітарного лікаря України від 10 грудня 1998 року № 7.

Мінімальні вимоги охорони праці під час роботи з дисплейними пристроями, незалежно від їх типу та моделі, визначені у "Вимогах щодо охорони праці та безпеки працівників під час роботи з дисплейними пристроями" (НПАОП 0.00-

7.15-18), затверджених Наказом Міністерства соціальної політики України № 207 від 14 лютого 2018 року.

Роботодавці повинні забезпечити проведення навчання і перевірки знань з питань охорони праці та безпечного використання екранних засобів, а також проведення медичних оглядів працівників перед початком роботи з екранними засобами (НПАОП 0.00-7.15-18).

Вимоги безпеки при використанні комп'ютерної техніки визначені у:

– ДСТУ EN 41003:2014 "Обладнання, підключене до телекомунікаційних мереж та/або кабельних розподільчих систем: Додаткові вимоги безпеки".

– ДСТУ EN 60335-1:2015 "Устаткування побутове та аналогічне електричне. Безпечність. Частина 1: Загальні вимоги".

– ДСТУ EN 60950-1:2015 "Апаратура інформаційно-обчислювальна. Безпечність. Частина 1: Загальні вимоги".

– ДСТУ EN 61140:2015 "Захист від ураження електричним струмом. Загальні аспекти встановлення та обладнання".

– ДСТУ EN 62368-1:2017 "Апаратура аудіо-, відео-, інформаційно-комунікаційна. Частина 1: Вимоги щодо безпеки".

Роботодавці, які використовують працю своїх працівників, повинні забезпечити комфортні та безпечні умови на робочому місці.

Робоча зона повинна бути розміром не менше шести квадратних метрів. За необхідності суміжні робочі місця працівників, які використовують комп'ютери, повинні бути відокремлені перегородками висотою до двох метрів; інші шафи, сейфи, тумби та інші меблі й обладнання, що знаходяться в приміщенні, також повинні бути враховані при визначенні достатнього розміру приміщення та робочої зони для однієї особи.

На робочому столі працівника може бути передбачено місце для допоміжного обладнання (принтер, колонка, сканер) та місце для зберігання документів, за умови, що видимість екрану не обмежується і не заважає працівнику.

Якщо рівень шуму та вібрації технічного обладнання є високим, роботодавець повинен забезпечити працівників антивібраційними килимками.

Робочі стільці повинні бути піднятими, легко регулюватися по висоті, забезпечувати належну підтримку спини та хребта, а також зручну позу. У приміщеннях має проводитися щоденне вологе прибирання, а робочі місця та комп'ютерні монітори повинні бути чистими від пилу. Компанії повинні чітко визначити перерви тривалістю 10-15 хвилин для відпочинку працівників (без урахування обіду), як правило, кожні одну-дві години, залежно від складності роботи. У будь-якому випадку, роботодавці повинні встановити в своїй компанії таку робочу програму, при якій тривалість безперервної роботи з комп'ютером не перевищувала б чотирьох годин.

Крім того, для підтримання належного рівня здоров'я та професійної придатності працівників рекомендується передбачити в компаніях окремі кімнати відпочинку, щоб працівники могли відпочити та зменшити нервове та емоційне напруження, викликане роботою за комп'ютером.

4.2 Безпека в надзвичайних ситуаціях. Дослідження стійкості роботи систем шифрування до впливу уражаючих факторів надзвичайних ситуацій воєнного часу

В умовах сучасного світу, де воєнні конфлікти є неминучою загрозою, стійкість роботи систем шифрування до впливу уражаючих факторів надзвичайних ситуацій воєнного часу є однією з найважливіших проблем інформаційної безпеки.

Шифрування є одним з основних засобів захисту інформації від несанкціонованого доступу. Воно використовується в різних сферах життєдіяльності, включаючи військову, державну, фінансову, промислову та інші.

У разі виникнення надзвичайної ситуації воєнного часу системи шифрування можуть піддаватися впливу різноманітних уражаючих факторів.

Вплив цих факторів може призвести до порушення роботи систем шифрування, що може створити загрозу для безпеки інформації, яка захищається.

Ефективність економіки держави в умовах воєнного часу залежить від того, наскільки системи шифрування, що використовуються для захисту інформації, здатні стійко працювати під впливом уражаючих факторів надзвичайних ситуацій.

Значні руйнування, пожежі та втрати серед населення, викликані наслідками НС, можуть призвести до порушення роботи систем шифрування, що може спричинити виток інформації, яка становить державну таємницю. Тому необхідно розробити заходи, спрямовані на забезпечення стійкості роботи систем шифрування до впливу уражаючих факторів НС.

На стійкість роботи системи на об’єктах впливають такі фактори:

– захищеність працівників від уражальних факторів у НС;

– здатність інженерно-технічного комплексу об’єкта (будівель, споруд, обладнання та комунально-енергетичних мереж) протистояти руйнівній дії уражальних факторів аварій, катастроф, стихійного лиха та сучасної зброї;

– надійність постачання об’єкта електроенергією, водою, паливом, комплектуючими та сировиною;

– підготовленість об’єкта до проведення аварійно-рятувальних та

– відновлюваних робіт;

– оперативність управління виробництвом та здійсненням заходів ЦЗ у НС.

Підвищення стійкості об’єкта досягають проведенням комплексу інженернотехнічних, технологічних, організаційних заходів.

До інженерно-технічних заходів належать роботи, що забезпечують стійкість виробничих будівель і споруд, обладнання та комунально-енергетичних систем. Технологічні заходи забезпечують підвищення стійкості об’єкта спрощенням технологічного процесу виробництва кінцевої продукції та виключенням або обмеженням розвитку аварій.

Організаційні заходи передбачають розробку ефективних дій керівного складу, служб та формувань ЦЗ, спрямованих на захист виробничого персоналу,

проведення рятувальних та інших невідкладних робіт, а також відновлення роботи об’єкту та його систем.

Заходи щодо підвищення стійкості об’єктів здійснюють відповідно до вимог Норм проектування інженерно-технічних заходів цивільного захисту. Дані вимоги призначені для того, щоб в умовах НС:

– забезпечити захист населення та знизити масштаби руйнувань (пожеж, затоплень, заражень);

– підвищити стійкість роботи об’єктів і галузей економіки;

– створити умови для успішного проведення робіт з ліквідації наслідків НС.

Вимоги Норм проектування реалізують під час проектування та забудови міст, будування нових промислових підприємств, об’єктів енергетики, транспортних систем, систем водо- та газопостачання, а також під час їх реконструкції.

Головним документом, відповідно до якого слід планувати та здійснювати інженерно-технічні заходи цивільного захисту є «Будівельні норми і правила» (БН і П 2.00.05-90), а також «Загальні вимоги до розвитку і розміщення потенційно небезпечних виробництв з урахуванням ризику надзвичайних ситуацій техногенного походження» (Київ, НАН України, 1995). Запровадження норм проектування ІТЗ ЦЗ здійснюється диференційовано з урахуванням ролі і важливості міст і об’єктів економіки. Для цього міста поділяють на групи, а об’єкти – на категорії за такою класифікацією: міста: «Особливої групи», I, II та III груп; об’єкти господарювання: «Особливої важливості», I та II категорій. Об’єкти атомної енергетики виділяють в окрему групу.

Перед тим, як планувати та вжити заходів щодо підвищення стійкості роботи будь-якого об’єкта, потрібно оцінити стійкість цього об’єкта. Мета оцінювання стійкості – виявлення найбільш слабких елементів виробництва відносно дій вражаючих факторів НС та розробка конкретних рекомендацій щодо підвищення стійкості як слабких елементів, так і об’єкта в цілому.

Для оцінювання реальної стійкості на об’єкті інженерно-технічний персонал об’єкта під керівництвом начальника ЦЗ (керівника підприємства) періодично проводить дослідження. На початковому етапі створюють дослідницькі групи, розробляють план досліджень та інші керівні документи.

Дослідницькі групи оцінюють стійкість інженерно-технічного комплексу, надійність захисту виробничого персоналу, стійкість постачання та управління за різних НС після попередньої підготовки.

Оцінювання стійкості об’єкта відбувається за такою методикою:

– оцінюють стійкість кожного елемента об’єкта;

– стійкість об’єкта в цілому визначають за стійкістю найбільш слабкого елемента.

– стійкість об’єкта оцінюють відносно кожного з можливих вражаючих факторів НС (варіантів аварій, стихійного лиха, застосування сучасної зброї);

– ураховують максимальні значення параметрів вражаючих факторів щодо умов розташування об’єкта.

Згідно навчального посібника «Техноекологія та цивільна безпека. Частина «Цивільна безпека» [5]: «Найбільш вразливими елементами систем енергопостачання є повітряні лінії електропередач, будівлі і споруди трансформаторних станцій та розподільчих пунктів. У разі руйнування будівель можливими є обриви проводів у середині приміщень, що при збереженні кабельних мереж призведе до короткого замикання, а ті в свою чергу можуть призвести до пожеж.

Для забезпечення надійного електропостачання в умовах НС при його проектуванні і будівництві також повинні враховуватися вимоги ІТЗ ЦЗ (ЦО). Електропостачання повинно здійснюватися від енергосистем, до складу яких входять електростанції, що працюють на різних видах палива. Великі електростанції потрібно розміщувати на значних відстанях одна від одної і від великих міст поза зоною можливих руйнувань. Районні понижувальні станції,

диспетчерські пункти та лінії електропередач необхідно розміщувати розосереджено і вони повинні бути надійно захищені.»

Науково-технічний прогрес характеризується зростанням кількості аварій, катастроф та посиленням їх руйнівного ефекту. Техногенні катастрофи мають таку періодичність або ймовірність: глобальні – 0,02–0,03 за рік; національні – 0,05–0,1 за рік; місцеві 1–20 за рік; об'єктові – 10–500 за рік.

На останнє десятиліття припадає майже половина загиблих і 40 % постраждалих у катастрофах під час стихійних лих XX століття. Вихід із такого становища один – зниження ризиків і пом'якшення наслідків НС, що вирішується на основі нової ідеології протидії катастрофам і розробленої на її базі державної стратегії управління ризиками.

В основу програми запобігання та реагування на НС техногенного та природного характеру покладено концепції прийнятного та виправданого ризику, стійкого розвитку суспільства. Концепцію прийнятного ризику використовують для раціонального планування заходів із забезпечення безпеки людей з урахуванням соціальних та економічних факторів. На її основі забезпечують техногенну безпеку. Прийнятний ризик – це ризик, який суспільство може забезпечити в певний період часу. Рівень прийнятного ризику встановлюється в державі законодавством.

За концепцією виправданого ризику прийнятний той ризик, котрий виправданий суспільством. При цьому представники суспільства, безпека яких на певному етапі розвитку науки і техніки не може бути забезпечена на прийнятому рівні (тих, хто реалізує нові технології з великим ризиком в інтересах суспільства), отримують соціально-економічні компенсації від суспільства.

Зниження ризиків і пом'якшення наслідків НС є стратегічним завданням держави у забезпеченні національної безпеки. У розв'язанні цього завдання важливе місце належить правовому забезпеченню.

Для розв'язання проблеми зниження ризику НС важливим є прогнозування і попередження аварій, катастроф, різних нестабільних ситуацій у природній і техногенній сферах.

Для своєчасного прогнозування і виявлення небезпечного природного явища на стадії його зародження потрібна добре налагоджена загальнодержавна система моніторингу за передвісниками стихійного лиха, катастрофи.

Методи прогнозування наслідків НС за часом проведення можна поділити на дві групи:

– що ґрунтуються на апріорних оцінках (припущеннях), отриманих за допомогою теоретичних моделей та аналогій;

– засновані на апостеріорних оцінках (оцінках наслідків НС, що вже трапилися).

Таким чином, прогнозування наслідків НС базується як на основі оцінки наслідків НС, що вже відбулися, так і припущеннях на основі моделей НС та аналогій.

ВИСНОВКИ

У цій кваліфікаційній роботі було досліджено залежність стійкості S-блоків до диференціального криптоаналізу від їх диференціальної рівномірності.

У цій кваліфікаційній роботі було проведено дослідження залежності стійкості S-блоків до диференціального криптоаналізу від їх диференціальної рівномірності. Для досягнення цієї мети було вирішено такі задачі:

– Розроблено алгоритм генерації S-блоків з заданим ступенем диференціальної рівномірності.

– Розроблено алгоритм диференціального криптоаналізу S-блоків в межах однораундового блокового шифру.

– Проведено диференціальний криптоаналіз S-блоків з різними ступенями диференціальної рівномірності.

За результатами дослідження було встановлено, що S-блоки з високою диференціальною рівномірністю є більш стійкими до диференціального криптоаналізу, ніж S-блоки з низькою диференціальною рівномірністю.

Це пояснюється тим, що S-блоки з високою диференціальною рівномірністю мають більш складну структуру, що ускладнює атакуючому використання диференціального криптоаналізу.

У рамках дослідження було проведено диференціальний криптоаналіз для множини S-блоків з різними ступенями диференціальної рівномірності.

Результати цього дослідження мають важливе значення для розробників блокових шифрів. Вони дозволяють їм краще зрозуміти, як диференціальна рівномірність S-блоків впливає на їхню стійкість до диференціального криптоаналізу.

Отримані в цій кваліфікаційній роботі результати є важливим внеском у дослідження стійкості S-блоків до диференціального криптоаналізу. Однак, існують і інші фактори, які можуть впливати на цю стійкість. Наприклад, важливу роль можуть відігравати такі фактори, як структура S-блоку, його взаємодія з

іншими компонентами шифру, а також методи диференціального криптоаналізу, які використовуються атакуючим.

У подальших дослідженнях необхідно вивчити вплив цих факторів на стійкість S-блоків до диференціального криптоаналізу. Це дозволить розробникам блокових шифрів поліпшити стійкість своїх шифрів до цього виду криптоаналізу.

СПИСОК ДЖЕРЕЛ

1. Загородна Н. В., Лупенко С. А., Луцків А. М. Сучасні алгебраїчні криптоаналітичні методи систем захисту мереж передачі даних. Інформаційні моделі, системи та технології : Матеріали тез доповідей I науково-технічної конференції, м. Тернопіль, 20 трав. 2011 р. Тернопіль, 2011. URL: http://elartu.tntu.edu.ua/handle/123456789/976.
2. Обґрунтування вимог, побудування та аналіз перспективних симетричних криптоперетворень на основі блочних шифрів / Кузнецов О. О та ін. Вісник Національного університету "Львівська політехніка". Комп’ютерні системи та мережі. 2014. № 806. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2017/nov/6634/21-124-141.pdf.
3. Оксьоненко М., Яковлєв С. Перевірка "випадковості" генерації S-блоків алгоритму шифрування ДСТУ 7624:2014. Київ : Національний технічний університет України "Київський політехнічний університет". URL: http://itcm.pnu.edu.ua/2016/docs/Oksonenko.pdf.
4. Піх В. В. Оцінка ефективності алгоритмів блоково-симетричного шифрування на основі використання міні-версій. Тернопіль, 2020. URL: http://elartu.tntu.edu.ua/handle/lib/33440.
5. Стручок В. С. Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2022. 150 с. URL: http://elartu.tntu.edu.ua/handle/lib/39424.
6. Ярема О. Практичні аспекти дослідження стійкості S-блоків до диференціального криптоаналізу. Інформаційні моделі, системи та технологіїї : Матеріали тез доповідей XI науково-технічної конференції, м. Тернопіль, 13–14 груд. 2023 р. Тернопіль, 2023. С. 135–136.
7. AL-Wattar A. H. S. A review of block cipher’s s-boxes tests criteria. Iraqi journal of statistical sciences. 2019. Vol. 16, no. 2. URL: https://doi.org/10.33899/iqjoss.2019.164195.

8. An STP-based model toward designing S-boxes with good cryptographic properties / Z. Lu et al. Designs, codes and cryptography. 2022. URL: https://doi.org/10.1007/s10623-022-01034-2 (date of access: 20.12.2023).
9. Borst J. Block ciphers: design, analysis and side-channel analysis. Belgium, 2011.
10. Cheung J. M. The design of S-boxes : Thesis. 2010. URL: http://hdl.handle.net/20.500.11929/sdsu:4689.
11. Eid Khamees Al-Shammary M., Mahmood Al-Dabbagh S. S. Differential Distribution Table implementation DDT survey. Technium: romanian journal of applied sciences and technology. 2022. Vol. 4, no. 10. P. 15–30. URL: https://doi.org/10.47577/technium.v4i10.7700 (date of access: 20.12.2023).
12. Freyre-Echevarría A., Martínez-Díaz I. On the construction of S-boxes using the leaders and followers metaheuristic. Cryptology ePrint Archive, Paper 2019/288, 2019. URL: https://eprint.iacr.org/2019/288.
13. Haci Ali Sahin. S-box classification and selection in symmetric-key algorithms. 2016. URL: https://hdl.handle.net/11511/25979.
14. Indesteege S., Preneel B. Practical Collisions for EnRUPT. Fast software encryption. Berlin, Heidelberg, 2009. P. 246–259. URL: https://doi.org/10.1007/978-3-642-03317-9_15 (date of access: 20.12.2023).
15. Jiménez R. A. d. l. C. On some methods for constructing almost optimal s-boxes and their resilience against side-channel attacks. IACR cryptol. eprint arch. 2018. Vol. 2018. URL: https://api.semanticscholar.org/CorpusID:51801605.
16. Kazlauskas K., Kazlauskas J. Key-Dependent s-box generation in AES block cipher system. Informatica. 2009. Vol. 20, no. 1. P. 23–34. URL: https://doi.org/10.15388/informatica.2009.235 (date of access: 20.12.2023).
17. Kruppa H., Shahy S. U. A. Differential and linear cryptanalysis in evaluating AES candidate algorithms : Technical report. National Institute of Standards and Technology. URL: https://www.cs.cmu.edu/~hannes/diffLinAES.pdf.
18. Marochok S., Zajac P. Algorithm for generating s-boxes with prescribed differential properties. Algorithms. 2023. Vol. 16, no. 3. P. 157. URL: https://doi.org/10.3390/a16030157 (date of access: 20.12.2023).

19. Nakahara J´unior J. Cryptanalysis and design of block ciphers : Ph. D. Leuven, 2003.
20. Results of Ukrainian national public cryptographic competition / V. Dolgov et al. Tatra mountains mathematical publications. 2012. Vol. 47, no. 1. URL: https://doi.org/10.2478/v10127-010-0033-6.
21. Sengel O., Aydin M. A., Sertbas A. An efficient generation and security analysis of substitution box using fingerprint patterns. IEEE access. 2020. Vol. 8. P. 160158–160176. URL: https://doi.org/10.1109/access.2020.3021055 (date of access: 20.12.2023).
22. Ullrich M. The design and efficient software implementation of s-boxes. Leuven, 2010. URL: https://mouha.be/wp-content/uploads/ullrich-thesis.pdf.
23. Wright R. N. Cryptography. Encyclopedia of physical science and technology. 2003. P. 61–77. URL: https://doi.org/10.1016/b0-12-227410-5/00843-7 (date of access: 20.12.2023).

ДОДАТОК А – АПРОБАЦІЯ НАУКОВОЇ РОБОТИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 001
М34

ПРОГРАМНИЙ КОМІТЕТ

Голова: Приймак Микола – професор кафедри комп'ютерних систем та мереж, д.т.н., професор.

Співголови: Марущак Павло – проректор з наукової роботи, докт. техн. наук, професор.
Баран Ігор – канд. техн. наук, доцент, декан факультету ФІС.
Науковий секретар: Семенишин Галина – старший викладач.

Члени: Василь Кривень - завідувач кафедри математичних методів в інженерії д.ф.-м.н., професор Галина Осухівська – завідувач кафедри комп'ютерних систем та мереж, к.т.н., доцент Микола Карпінський - професор кафедри кібербезпеки, д.т.н., професор Жанна Баб'як - завідувач кафедри української та іноземних мов, к.пед. н., доцент Ярослав Литвиненко – професор кафедри комп'ютерних наук, д.т.н., професор Михайло Петрик - завідувач кафедри програмної іженерії, д.ф.-м.н., професор Наталія Загородна – завідувач кафедри кібербезпеки, к.т.н., доцент.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова: Скоренький Юрій Любомирович – канд. техн. наук, доцент кафедри фізики.
Члени: доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; доцент кафедри кібербезпеки, к.т.н. М. Стадник; асистент Н. Шаблій; ст. викладач Л. Джиджора.

М34 Матеріали XI науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 13-14 грудня 2023 р.). – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2023. – 271 с.

Адреса оргкомітету: ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001,
тел. (0352) 52-41-33, факс (0352) 254983.
E-mail: **conffis2023@gmail.com**

Редагування, оформлення, верстка: Семенишин Г.М.

СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання;
- Інформаційні системи та технології;
- Комп'ютерні системи та мережі;
- Програмна інженерія та моделювання складних розподілених систем;
- Новітні фізико-технічні та освітні технології.

В збірнику надруковано тези доповідей XI науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 13-14 грудня 2023 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор.

УДК 004.056.55
О. Ярема
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ПРАКТИЧНІ АСПЕКТИ ДОСЛІДЖЕННЯ СТІЙКОСТІ S-БЛОКІВ ДО ДИФЕРЕНЦІАЛЬНОГО КРИПТОАНАЛІЗУ

O. Yarema

PRACTICAL ASPECTS OF RESEARCH ON THE RESISTANCE OF S-BLOCKS TO DIFFERENTIAL CRYPTANALYSIS

S-блоки (блоки підстановки) є ключовим елементом багатьох блокових шифрів, таких як DES та AES. Вони використовуються для забезпечення нелінійності в процесі шифрування, що є критично важливим для забезпечення стійкості шифру до різних видів криптоаналітичних атак, як статистичних, так і алгебраїчних. S-блоки є статичними для конкретної реалізації блокового шифру з деякими винятками [1] і не залежать від секретного ключа, що робить їх одним з головних об'єктів атаки на шифр. S-блоки можуть задаватися таблично (див. рисунок 1) або як набір інструкцій алгебраїчних перетворень для вхідних бітів.

	00	01	02	03	04	05	06	07	08
00	63	7c	77	7b	f2	6b	6f	c5	30
10	ca	82	c9	7d	fa	59	47	f0	ad
20	b7	fd	93	26	36	3f	f7	cc	34
30	04	c7	23	c3	18	96	05	9a	07
40	09	83	2c	1a	1b	6e	5a	a0	52
50	53	d1	00	ed	20	fc	b1	5b	6a
60	d0	ef	aa	fb	43	4d	33	85	45
70	51	a3	40	8f	92	9d	38	f5	bc
80	cd	0c	13	ec	5f	97	44	17	c4

Рисунок 1. Частина табличного задання S-блока шифру AES

Визначення, чи підходить конкретний S-блок для використання в шифрі, відбувається на основі дослідження основних криптографічних властивостей S-блоків: критерію лавини, нелінійності, незалежності бітів, диференційної рівномірності, зворотності та несуперечності.

Критерій лавини є мірою того, наскільки ефективно S-блок розповсюджує вхідні зміни на вихідні біти. Якщо зміна одного біту на вході призводить до зміни багатьох бітів на виході, то критерій лавини вважається виконаним [2]. Більш стійкою характеристикою блоків підстановки є строгий критерій лавини.

Нелінійність S-блоку забезпечує, що висока лінійна кореляція між вхідними та вихідними бітами буде мінімальною, що робить лінійний криптоаналіз менш ефективним [3].

Незалежність бітів означає, що зміна одного біту вхідної інформації не повинна мати вплив на інші біти вихідної інформації. Ця властивість демонструє, що кожен біт вихідної інформації може бути обчислений незалежно від інших. По суті, ця властивість характеризує те ж явище, що й критерій лавини, але з іншої точки зору [4].

Диференційна рівномірність вказує на те, як часто можливий певний диференціал між двома блоками даних. Чим менше ймовірність того, що можливий диференціал зустріне, тим більша стійкість шифру до диференціальних атак.

Зворотність та несуперечність це пов'язана пара властивостей, які описують те, чи для кожного вхідного значення існує лише одне відповідне вихідне значення і навпаки.

Література

1. Agarwal P., Singh A., Kilicman A. Development of key-dependent dynamic S-Boxes with dynamic irreducible polynomial and affine constant. Advances in mechanical engineering. 2018. Т. 10, № 7. С. 168781401878163. URL: https://doi.org/10.1177/1687814018781638 (дата звернення: 25.11.2023).
2. Kim K., Matsumoto T., Imai H. A recursive construction method of s-boxes satisfying strict avalanche criterion. Advances in Cryptology-CRYPT0' 90. Berlin, Heidelberg. С. 565–574. URL: https://doi.org/10.1007/3-540-38424-3_39 (дата звернення: 25.11.2023).
3. Nyberg K. S-boxes and round functions with controllable linearity and differential uniformity. Fast software encryption. Berlin, Heidelberg, 1995. С. 111–130. URL: https://doi.org/10.1007/3-540-60590-8_9 (дата звернення: 25.11.2023).
4. Sinha S., Arya C. Algebraic construction and cryptographic properties of rijndael substitution box. Defence science journal. 2012. Т. 62, № 1. С. 32–37. URL: https://doi.org/10.14429/dsj.62.1439 (дата звернення: 25.11.2023).

ДОДАТОК Б – ЛІСТИНГИ КОДУ

Лістинг 1 – Генератор S-блоків

```
from random import randrange
import differential_table
BITS = 5
def generate_sbox(bits=BITS, console_print=False):
    max_value = 2 ** bits
    bits_list = [i for i in range(max_value)]
    s_box = {}
    for i in range(max_value):
        ind = randrange(0, len(bits_list))
        s_box[i] = bits_list.pop(ind)
    if console_print:
        for key, value in s_box.items():
            print(f"{key}: {value}, ", end="")
    return s_box
def check_diff_uniformity(s_box, bits=BITS):
    (ddt, ddt_input_pairs) =
differential_table.generate_dd_table(s_box, bits)
    values = []
    for i in range(len(ddt)):
        for j in range(len(ddt[i])):
            if ddt[i][j] != "-":
                values.append(ddt[i][j])
    values.remove(max(values))
    return max(values), ddt, ddt_input_pairs
```

Лістинг 2 – Знаходження таблиці диференціального розподілу

```
# CONFIG ##################################################
# s-box of the cipher
S_BOX = {0: 11, 1: 19, 2: 17, 3: 27, 4: 29, 5: 26, 6: 8, 7: 25, 8:
0, 9: 1, 10: 5, 11: 31, 12: 28, 13: 21, 14: 10, 15: 9, 16: 2}
```

Продовження лістингу 2

```
# number of bits in plain, ciphered text or half of the key
BITS = 5
#################################################################
def generate_dd_table(s_box=S_BOX, bits=BITS, console_print=False):
    """Generate DDT and all the input pairs that DDT consist of in
form of two two-dimensional lists."""
    max_value = 2 ** bits
    # set up empty table
    table = [["-" for j in range(max_value)] for i in
range(max_value)]
    table_of_pairs = [[[] for j in range(max_value)] for i in
range(max_value)]
    # check XOR difference for every possible pair of input
    for i in range(max_value):
        for j in range(max_value):
            input_dif = i ^ j
            output_diff = s_box[i] ^ s_box[j]
            if table[output_diff][input_dif] == "-":
                table[output_diff][input_dif] = 0
            table[output_diff][input_dif] += 1
            table_of_pairs[output_diff][input_dif].append((i, j))
            print(f"Progress: {(i*max_value+j+1)/(max_value *
max_value)*100}%")
    return table, table_of_pairs
```